

日志分析场景公开

access_log数据集

[ClickHouse表结构DDL](#)

[Elasticsearch表结构DDL](#)

[ClickHouse查询语句](#)

[Elasticsearch查询语句](#)

trace_log数据集

[ClickHouse表结构DDL](#)

[Elasticsearch表结构DDL](#)

[ClickHouse查询语句](#)

[Elasticsearch查询语句](#)

access_log数据集

ClickHouse表结构DDL

```
1 CREATE TABLE access_log_local ON CLUSTER default
2 (
3     `sql` String,
4     `schema` String,
5     `type` String,
6     `access_ip` String,
7     `conn_id` UInt32,
8     `process_id` String,
9     `logic_ins_id` UInt32,
10    `accept_time` UInt64,
11    `_date` DateTime,
12    `total_time` UInt32,
13    `succeed` String,
14    `inst_name` String
15 )
16 ENGINE = MergeTree()
17 PARTITION BY toYYYYMMDD(_date)
```

```
18 ORDER BY (logic_ins_id, accept_time);
19
20 CREATE TABLE access_log on cluster default as access_log_local
21 engine = Distributed(default, default, access_log_local, rand());
```

Elasticsearch表结构DDL

```
1 PUT /access_log?pretty
2 {
3   "settings": {
4     "index": {
5       "number_of_shards": 8,
6       "number_of_replicas": 1,
7       "max_result_window": 1000000,
8       "max_inner_result_window": 10000,
9       "requests.cache.enable": false,
10      "queries.cache.enabled": false,
11      "warmer.enabled": false,
12      "sort.field": [
13        "logic_ins_id",
14        "accept_time"
15      ]
16    }
17  },
18  "mappings": {
19    "properties": {
20      "_date": {
21        "type": "date",
22        "format": "yyyy-MM-dd HH:mm:ss||yyyy-MM-dd||epoch_millis"
23      },
24      "accept_time": {
25        "type": "long"
26      },
27      "access_ip": {
28        "type": "keyword"
29      },
30      "conn_id": {
31        "type": "integer"
```

```

32     },
33     "inst_name": {
34         "type": "keyword"
35     },
36     "logic_ins_id": {
37         "type": "integer"
38     },
39     "process_id": {
40         "type": "keyword"
41     },
42     "schema": {
43         "type": "keyword"
44     },
45     "sql": {
46         "type": "text"
47     },
48     "succeed": {
49         "type": "keyword"
50     },
51     "total_time": {
52         "type": "integer"
53     },
54     "type": {
55         "type": "keyword"
56     }
57 }
58 }
59 }

```

ClickHouse查询语句

```

1 --Q1
2 select _date, accept_time, access_ip, type, total_time, concat(to
String(total_time),'ms') as total_time_ms, sql,schema,succeed,pro
cess_id,inst_name from access_log where _date >= '2020-12-27 00:3
8:31' and _date <= '2020-12-28 00:38:31' and logic_ins_id = 50268
0264 and accept_time <= 1609087111000 and accept_time >= 16090007

```

```

11000 and positionCaseInsensitive(sql, 'select') > 0 order by acc
ept_time desc limit 50,50;
3 --Q2
4 select
5 case
6 when total_time <=100 then 1
7 when total_time > 100 and total_time <= 500 then 2
8 when total_time > 500 and total_time <= 1000 then 3
9 when total_time > 1000 and total_time <= 3000 then 4
10 when total_time > 3000 and total_time <= 10000 then 5
11 when total_time > 10000 and total_time <= 30000 then 6
12 else 7
13 end as reorder,
14 case
15 when total_time <=100 then '0~100ms'
16 when total_time > 100 and total_time <= 500 then '100ms~500ms'
17 when total_time > 500 and total_time <= 1000 then '500ms~1s'
18 when total_time > 1000 and total_time <= 3000 then '1s~3s'
19 when total_time > 3000 and total_time <= 10000 then '3s~10s'
20 when total_time > 10000 and total_time <= 30000 then '10s~30s'
21 else '30s以上'
22 end as label,
23 case
24 when total_time <= 100 then '0~100'
25 when total_time > 100 and total_time <= 500 then '100~500'
26 when total_time > 500 and total_time <= 1000 then '500~1000'
27 when total_time > 1000 and total_time <= 3000 then '1000~3000'
28 when total_time > 3000 and total_time <= 10000 then '3000~10000'
29 when total_time > 10000 and total_time <= 30000 then '10000~3000
0'
30 else '30000~100000000000'
31 end as vlabel,
32 count() as value
33 from access_log
34 where logic_ins_id = 502867976 and _date >= '2020-12-27 00:38:31'
and _date <= '2020-12-28 00:38:31' and accept_time <= 16090871110
00 and accept_time >= 1609000711000
35 group by label,vlabel,reorder
36 order by reorder;
37 --Q3

```

```

38 select toStartOfMinute(_date) as time, count() as value
39 from access_log
40 where logic_ins_id = 500152868 and accept_time <= 1609087111000 a
   nd accept_time >= 1609000711000
41 group by time
42 order by time;
43 --Q4
44 select count(*) as c from (
45   select _date, accept_time, access_ip, type, total_time, concat(
   toString(total_time),'ms') as total_time_ms, sql, schema, succeed
   , process_id, inst_name
46   from access_log
47   where logic_ins_id = 501422856 and _date >= '2020-12-27 00:38:3
   1' and _date <= '2020-12-28 00:38:31' and accept_time <= 16090871
   11000 and accept_time >= 1609000711000
48 );

```

Elasticsearch查询语句

```

1 --Q1
2 POST /_sql?format=txt&pretty
3 {
4   "query": "select _date, accept_time, access_ip, type, total_tim
   e, concat(cast(total_time as string),'ms') as total_time_ms, sql,
   schema,succeed,process_id,inst_name from access_log where _date >
   = '2020-12-27 00:38:31' and _date <= '2020-12-28 00:38:31' and lo
   gic_ins_id = 502680264 and accept_time <= 1609087111000 and accep
   t_time >= 1609000711000 and match(sql, 'select') order by accept_
   time desc limit 100"
5 }
6
7 --Q2
8 POST /_sql?format=txt&pretty
9 {
10  "query": "select case when total_time <=100 then 1 when total_t
   ime > 100 and total_time <= 500 then 2 when total_time > 500 and
   total_time <= 1000 then 3 when total_time > 1000 and total_time
   <= 3000 then 4 when total_time > 3000 and total_time <= 10000 th

```

```

en 5 when total_time > 10000 and total_time <= 30000 then 6 else
  7 end as reorder, case when total_time <=100 then '0~100ms' when
total_time > 100 and total_time <= 500 then '100ms~500ms' when to
tal_time > 500 and total_time <= 1000 then '500ms~1s' when total_
time > 1000 and total_time <= 3000 then '1s~3s' when total_time >
3000 and total_time <= 10000 then '3s~10s' when total_time > 1000
0 and total_time <= 30000 then '10s~30s' else '30s以上' end as lab
el, case when total_time <= 100 then '0~100' when total_time > 10
0 and total_time <= 500 then '100~500' when total_time > 500 and
total_time <= 1000 then '500~1000' when total_time > 1000 and to
tal_time <= 3000 then '1000~3000' when total_time > 3000 and tota
l_time <= 10000 then '3000~10000' when total_time > 10000 and tot
al_time <= 30000 then '10000~30000' else '30000~100000000000' end
as vlabel, count(*) as value from access_log where logic_ins_id
= 502867976 and _date >= '2020-12-27 00:38:31' and _date <= '202
0-12-28 00:38:31' and accept_time <= 1609087111000 and accept_tim
e >= 1609000711000 group by label,vlabel,reorder order by reorde
r"
11 }
12
13 --Q3
14 POST /_sql?format=txt&pretty
15 {
16   "query": "select date_trunc('mi', _date) as time, count(*) as v
value from access_log where logic_ins_id = 500152868 and accept_ti
me <= 1609087111000 and accept_time >= 1609000711000 group by 1 o
rder by 1"
17 }
18
19 --Q4
20 POST /_sql?format=txt&pretty
21 {
22   "query": "select count(*) as c from (select _date, accept_time,
access_ip, type, total_time, concat(cast(total_time as string),'m
s') as total_time_ms, sql, schema, succeed, process_id, inst_name
from access_log where logic_ins_id = 501422856 and _date >= '2020
-12-27 00:38:31' and _date <= '2020-12-28 00:38:31' and accept_ti
me <= 1609087111000 and accept_time >= 1609000711000)"
23 }

```

trace_log数据集

ClickHouse表结构DDL

```
1 CREATE TABLE trace_local on cluster default
2 (
3   `serviceName` LowCardinality(String),
4   `host` LowCardinality(String),
5   `ip` String,
6   `spanName` String,
7   `spanId` String,
8   `pid` LowCardinality(String),
9   `parentSpanId` String,
10  `ppid` String,
11  `duration` Int64,
12  `rpcType` Int32,
13  `startTime` Int64,
14  `traceId` String,
15  `tags.k` Array(String),
16  `tags.v` Array(String),
17  `events` String,
18  KEY trace_idx traceId TYPE range
19 ) ENGINE = MergeTree()
20 PARTITION BY intDiv(startTime, toInt64(7200000000))
21 PRIMARY KEY (serviceName, host, ip, pid, spanName)
22 ORDER BY (serviceName, host, ip, pid, spanName, tags.k);
23
24 CREATE TABLE trace on cluster default as trace_local
25 engine = Distributed(default, default, trace_local, rand());
```

Elasticsearch表结构DDL

```
1 PUT /trace?pretty
2 {
3   "settings" : {
4     "index" : {
5       "number_of_shards" : 32,
```

```

6     "number_of_replicas" : 1,
7     "max_result_window" : 1000000,
8     "max_inner_result_window" : 10000,
9     "requests.cache.enable": false,
10    "queries.cache.enabled": false,
11    "warmer.enabled": false,
12    "sort.field": [ "serviceName", "host", "ip", "pid", "spanName", "tags.k" ]
13  }
14 },
15 "mappings" : {
16   "properties" : {
17     "duration" : {
18       "type" : "long"
19     },
20     "events" : {
21       "type" : "keyword"
22     },
23     "host" : {
24       "type" : "keyword"
25     },
26     "ip" : {
27       "type" : "keyword"
28     },
29     "parentSpanId" : {
30       "type" : "keyword"
31     },
32     "pid" : {
33       "type" : "keyword"
34     },
35     "ppid" : {
36       "type" : "keyword"
37     },
38     "rpcType" : {
39       "type" : "integer"
40     },
41     "serviceName" : {
42       "type" : "keyword"
43     },
44     "spanId" : {

```



```

45     "type" : "keyword"
46   },
47   "spanName" : {
48     "type" : "keyword"
49   },
50   "startTime" : {
51     "type" : "long"
52   },
53   "tags" : {
54     "properties" : {
55       "k" : {
56         "type" : "keyword"
57       },
58       "v" : {
59         "type" : "keyword"
60       }
61     }
62   },
63   "traceId" : {
64     "type" : "keyword"
65   }
66 }
67 }
68 }

```

ClickHouse查询语句

```

1 --Q1
2 select *
3 from trace
4 prewhere
5 traceId = 'ccc6084420b76183'
6 where startTime > 1597968000300000 and startTime < 159805439909
  9000 settings max_threads = 1;
7 --Q2
8 select count(*) count, spanName as name from trace
9 where serviceName = 'conan-dean-user-period'
10 and startTime > 1597968000300000 and startTime < 15980543990990

```

```

00
11 group by spanName
12 order by count desc limit 1000;
13 --Q3
14 select host as name, count(*) count
15 from trace
16 where serviceName = 'conan-dean-user-period'
17 and startTime > 1597968000300000 and startTime < 15980543990990
00
18 group by host;
19 --Q4
20 select count(*) count, tags.k as name from trace
21 array join tags.k
22 where serviceName = 'conan-dean-user-period'
23 and startTime > 1597968000300000 and startTime < 15980543990990
00
24 group by tags.k;
25 --Q5
26 select count(*) spancount,
27 sum(duration) as sumDuration, intDiv(startTime, 144000000) as ti
    meSel
28 from trace
29 where serviceName = 'conan-dean-user-period'
30 and startTime > 1597968000300000 and startTime < 15980543990990
00
31 group by timeSel;
32 --Q6
33 select count(*) spanCount,
34 countIf(duration <=1000000), countIf(duration > 1000000), count
    If(duration > 3000000)
35 from trace
36 where serviceName = 'conan-dean-user-period'
37 and startTime > 1597968000300000 and startTime < 15980543990990
00;
38 --Q7
39 select host, startTime,traceId,spanName,duration,tags.k,tags.v
40 from trace
41 where serviceName = 'conan-dean-user-period'
42 and startTime > 1597968000300000 and startTime < 15980543990990
00 limit 1000000;

```

Elasticsearch查询语句

```
1 --Q1
2 GET /trace/_search?pretty
3 {
4   "size" : 1000,
5   "query" : {
6     "bool" : {
7       "must" : [
8         {
9           "term" : {
10            "traceId" : {
11              "value" : "ccc6084420b76183",
12              "boost" : 1.0
13            }
14          }
15        },
16        {
17          "range" : {
18            "startTime" : {
19              "from" : 1597968000300000,
20              "to" : 1598054399099000,
21              "include_lower" : false,
22              "include_upper" : false,
23              "time_zone" : "Z",
24              "boost" : 1.0
25            }
26          }
27        }
28      ],
29      "adjust_pure_negative" : true,
30      "boost" : 1.0
31    }
32  },
33  "sort" : [
34    {
35      "_doc" : {
```

```

36     "order" : "asc"
37   }
38 }
39 ]
40 }
41
42 --Q2
43 POST /_sql?format=txt&pretty
44 {
45   "query": "select count(*) count, spanName as name from trace w
here serviceName ='conan-dean-user-period' and startTime > 15979
68000300000 and startTime < 1598054399099000 group by spanName
order by count desc limit 1000"
46 }
47
48 --Q3
49 POST /_sql?format=txt&pretty
50 {
51   "query": "select host as name, count(*) count from trace where
serviceName ='conan-dean-user-period' and startTime > 1597968000
300000 and startTime < 1598054399099000 group by host"
52 }
53
54 --Q4
55 POST /_sql?format=txt&pretty
56 {
57   "query": "select count(*) count, tags.k as name from trace wh
ere serviceName ='conan-dean-user-period' and startTime > 159796
8000300000 and startTime < 1598054399099000 group by tags.k"
58 }
59
60 --Q5
61 POST /_sql?format=txt&pretty
62 {
63   "query": "select count(*) spancount, sum(duration) as sumDurat
ion, (startTime / 1440000000) as timeSel from trace where servic
eName ='conan-dean-user-period' and startTime > 1597968000300000
and startTime < 1598054399099000 group by timeSel"
64 }
65

```

```
66 --Q6
67 GET /trace/_search
68 {
69   "size" : 0,
70   "query" : {
71     "bool" : {
72       "must" : [
73         {
74           "term" : {
75             "serviceName" : {
76               "value" : "conan-dean-user-period",
77               "boost" : 1.0
78             }
79           }
80         },
81         {
82           "range" : {
83             "startTime" : {
84               "from" : 1597968000300000,
85               "to" : 1598054399099000,
86               "include_lower" : false,
87               "include_upper" : false,
88               "time_zone" : "Z",
89               "boost" : 1.0
90             }
91           }
92         }
93       ],
94       "adjust_pure_negative" : true,
95       "boost" : 1.0
96     }
97   },
98   "_source" : false,
99   "stored_fields" : "_none_",
100  "track_total_hits" : 2147483647,
101  "aggregations" : {
102    "groupby" : {
103      "filters" : {
104        "filters" : [
105          {
```

```

106         "match_all" : {
107             "boost" : 1.0
108         }
109     },
110 ],
111     "other_bucket" : false,
112     "other_bucket_key" : "_other_"
113 },
114 "aggregations" : {
115     "2227391e" : {
116         "sum" : {
117             "script" : {
118                 "source" : "InternalSqlScriptUtils.cast(InternalQl
ScriptUtils.lte(InternalQlScriptUtils.docValue(doc,params.v0),pa
rams.v1),params.v2)",
119                 "lang" : "painless",
120                 "params" : {
121                     "v0" : "duration",
122                     "v1" : 1000000,
123                     "v2" : "INTEGER"
124                 }
125             }
126         }
127     },
128     "22265801" : {
129         "sum" : {
130             "script" : {
131                 "source" : "InternalSqlScriptUtils.cast(InternalQl
ScriptUtils.gt(InternalQlScriptUtils.docValue(doc,params.v0),par
ams.v1),params.v2)",
132                 "lang" : "painless",
133                 "params" : {
134                     "v0" : "duration",
135                     "v1" : 1000000,
136                     "v2" : "INTEGER"
137                 }
138             }
139         }
140     },
141     "1838381" : {

```

```

142     "sum" : {
143         "script" : {
144             "source" : "InternalSqlScriptUtils.cast(InternalQl
ScriptUtils.gt(InternalQlScriptUtils.docValue(doc,params.v0),par
ams.v1),params.v2)",
145             "lang" : "painless",
146             "params" : {
147                 "v0" : "duration",
148                 "v1" : 3000000,
149                 "v2" : "INTEGER"
150             }
151         }
152     }
153 }
154 }
155 }
156 }
157 }
158
159 --Q7
160 GET /trace/_search
161 {
162     "size": 1000000,
163     "query": {
164         "bool": {
165             "must": [
166                 {
167                     "term": {
168                         "serviceName": {
169                             "value": "conan-dean-user-period",
170                             "boost": 1
171                         }
172                     }
173                 },
174                 {
175                     "range": {
176                         "startTime": {
177                             "from": 1597968000300000,
178                             "to": 1598054399099000,
179                             "include_lower": false,

```

```
180         "include_upper": false,
181         "time_zone": "Z",
182         "boost": 1
183     }
184 }
185 }
186 ],
187     "adjust_pure_negative": true,
188     "boost": 1
189 }
190 },
191 "_source": {
192     "includes": [
193         "host",
194         "startTime",
195         "traceId",
196         "spanName",
197         "duration",
198         "tags.k",
199         "tags.v"
200     ],
201     "excludes": []
202 },
203 "sort": [
204     {
205         "_doc": {
206             "order": "asc"
207         }
208     }
209 ]
210 }
```