实现一种像google搜索一样方便的es查询api

黄琛(C.Wong)

https://github.com/huangchen007/elasticsearch-rest-command

elasticsearch的官方query dsl

```
{
    "match" : {
        "message" : {
            "query" : "this is a test",
            "operator" : "and"
        }
    }
}
```

elasticsearch的官方api特点

优点：

直观，可读性强

机器解析速度也快

扩展性好

缺点：

编写复杂，标点符号多

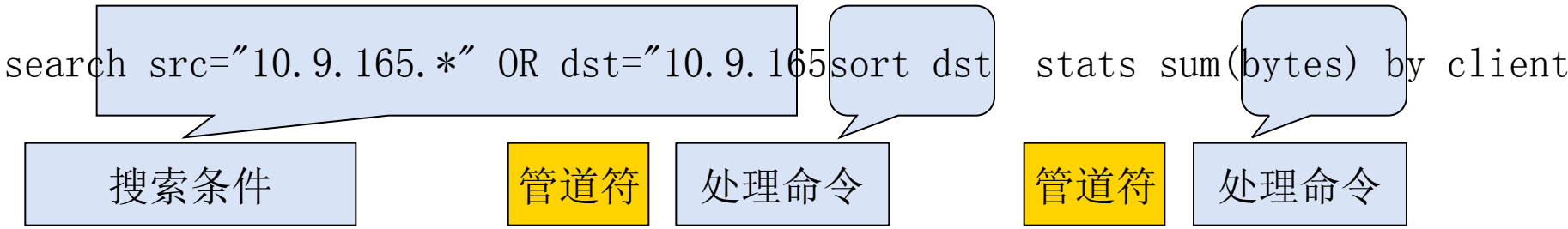
学习成本稍高

目前更加流行的人机交互接口

- SQL系列
- grep/awk
- google搜索/高级搜索

这样的接口

- search src="10.9.165.*" OR dst="10.9.165.8"
- search src="10.9.165.*" OR dst="10.9.165.8" | sort dst
- search src="10.9.165.*" OR dst="10.9.165.8" | sort dst | stats sum(bytes) as ASumOfBytes by clientip



支持的命令

- search(2014-4-1)
  - search <searchoptions> <BooleanExpression>
  - logicalexpression:and/or/not/嵌套/default=and
  - operator:[!]=/>[=]/<[=]/
  - searchoptions: sourcetype/index/hasparent/haschild
- sort (2014-4-21)
  - ...| sort <field> [DESC/ASC]
- stats (2014-4-10)
  - ... | stats [statsoptions] <StatsFunction> [ByClause]
  - StatsFunction:count(..),sum(..),avg(..),max(..),min(..),dc(...)
  - parameter:fieldname/eval(script)
  - ByClause: By <parameterlist>
  - statsoptions:minicount,limit,span,timespan
- join (2014-06-13)
  - ... | join <fieldlist> (subsearch)
  - eg:index=comment | join userid (search index=user)
- table(2014-06-16)
  - ... | table <filedlist>
  - filedlist支持通配符

实现/设计

- es插件实现vs直接修改代码
- javacc vs antlr

程序包

- com.everdata.command
  - 根据语法解析的结果对es集群进行查询并返回结果
- com.everdata.parser
  - 语法解析，转换成query dsl
- org.elasticsearch.plugin.rest
  - 实现plugin所必要的接口

主要调用流程

```
class CommandRestHandler extends BaseRestHandler{
    void handleRequest(...){
        commandString = request.param("q", "");
        parser = new CommandParser(commandString);
        search = new Search(parser, client, logger);
        switch(mode){
            case 0:result = search.executeQueryWithNonJoin();
            case 1:result = search.executeQuery();
            case 2:result = search.executeReport();
            case 3:result = search.executeDownload();
            case 4:result = search.executeDelete()
        }
    }
}
```

程序包org.elasticsearch.plugin.rest的分层结构

类分层结构

- java.lang.**Object**
  - org.elasticsearch.common.component.AbstractComponent
    - org.elasticsearch.rest.BaseRestHandler (implements org.elasticsearch.rest.RestHandler)
      - org.elasticsearch.plugin.rest.**CommandRestHandler**
      - org.elasticsearch.plugin.rest.**JobRestHandler**
  - org.elasticsearch.common.inject.AbstractModule (implements org.elasticsearch.common.inject.Module)
    - org.elasticsearch.plugin.rest.**CommandRestModule**
  - org.elasticsearch.plugins.AbstractPlugin (implements org.elasticsearch.plugins.Plugin)
    - org.elasticsearch.plugin.rest.**CommandRestPlugin**

程序包com.everdata.command的分层结构

类分层结构

- java.lang.Object
  - com.everdata.command.Field
  - com.everdata.command.Function
  - com.everdata.command.JoinQuery
  - com.everdata.command.JoinQuery.Join
  - com.everdata.command.Option
  - com.everdata.command.ReportResponse
  - com.everdata.command.Search
  - com.everdata.command.Search.QueryResponse
  - java.lang.Throwable (implements java.io.Serializable)
    - java.lang.Exception
      - com.everdata.command.CommandException

程序包com.everdata.parser的分层结构

类分层结构

- java.lang.**Object**
  - com.everdata.parser.**AST_ByIdentList.By**
  - com.everdata.parser.**AST_Sort.SortField**
  - com.everdata.parser.**AST_Stats.Bucket**
  - com.everdata.parser.**CommandParser** (implements com.everdata.parser.CommandParserConstants, com.everdata.parse
  - com.everdata.parser.**CommandParserTokenManager** (implements com.everdata.parser.CommandParserConstants)
  - com.everdata.parser.**Expression**
  - com.everdata.parser.**JavaCharStream**
  - com.everdata.parser.**JJTCommandParserState**
  - com.everdata.parser.**SimpleCharStream**
  - com.everdata.parser.**SimpleNode** (implements com.everdata.parser.Node)
    - com.everdata.parser.**AST_AndExpr**
    - com.everdata.parser.**AST_ByIdentList**
    - com.everdata.parser.**AST_ComparisonExpression**
    - com.everdata.parser.**AST_Delete**
    - com.everdata.parser.**AST_EvalExpr**
    - com.everdata.parser.**AST_FieldExpr**
    - com.everdata.parser.**AST_IdentList**
    - com.everdata.parser.**AST_Join**
    - com.everdata.parser.**AST_OrExpr**
    - com.everdata.parser.**AST_PredicateExpression**
    - com.everdata.parser.**AST_Regex**
    - com.everdata.parser.**AST_Search**
    - com.everdata.parser.**AST_SearchOption**
    - com.everdata.parser.**AST_Sort**
    - com.everdata.parser.**AST_Start**
    - com.everdata.parser.**AST_Stats**
    - com.everdata.parser.**AST_StatsFunc**
    - com.everdata.parser.**AST_Table**
    - com.everdata.parser.**AST_TermExpression**
    - com.everdata.parser.**AST_TopOption**
    - com.everdata.parser.**AST_UnaryExpr**
  - java.lang.**Throwable** (implements java.io.Serializable)
    - java.lang.**Error**
      - com.everdata.parser.**TokenMgrError**
    - java.lang.**Exception**
      - com.everdata.parser.**ParseException**
  - com.everdata.parser.**Token** (implements java.io.Serializable)

如何表示嵌套的布尔表达式

SearchStatement::=<K_SEARCH> (SearchOption)* (BooleanExpression)?

BooleanExpression::=AndExpression (<K_OR> AndExpression)*

AndExpression::=UnaryExpression (<K_AND> UnaryExpression)*

UnaryExpression::=(<K_NOT>)? (<O_LPAREN> BooleanExpression <O_RPAREN> |
PredicateExpression)

PredicateExpression::=(ComparisonExpression (ComparisonExpression |
TermExpression)* | TermExpression (ComparisonExpression | TermExpression)*)

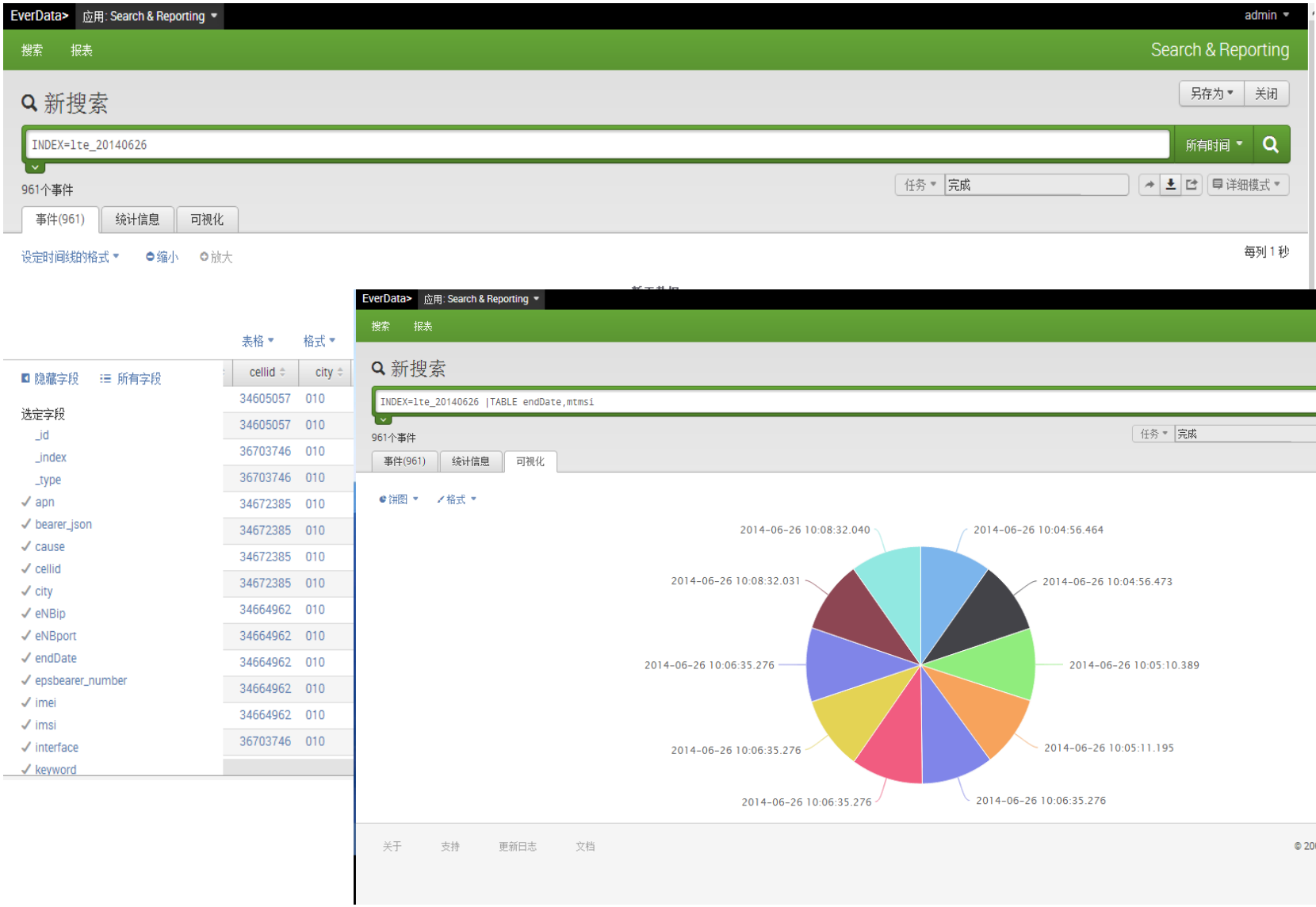
TermExpression::=((<S_INTEGER> | <S_FLOAT> | <S_IDENTIFIER>) |
<S_QUOTED_STRING>)

ComparisonExpression::=<S_IDENTIFIER> (<O_EQ> | <O_NEQ> | <O_GT> |
<O_GTE> | <O_LT> | <O_LTE>) ((<S_INTEGER> | <S_FLOAT> |
<S_IDENTIFIER>) | <S_QUOTED_STRING>)

递归遍历Abstract Syntax Tree得到Query DSL

```
QueryBuilder genQueryBuilder(SimpleNode tree){
   switch(nodeType){
   case JJT_OREXPR:
        return fb.or(genQueryBuilder(tree.children));
   case JJT_ANDEXPR/JJT_PREDICATEEXPRESSION:
        return fb.must(genQueryBuilder(tree.children));
   case JJT_UNARYEXPR:
        return fb.mustnot(genQueryBuilder(tree.children));
   case JJT_COMPARISONEXPRESSION/JJT_TERMEXPRESSION:
        return QueryBuilders.termQuery/matchPhraseQuery
        /rangeQuery/prefixQuery/wildcardQuery;
   }
}
```

UI（演示）


EverData>
应用: Search & Reporting
admin
搜索
报表
Search & Reporting
新搜索
另存为
关闭
INDEX=lte_20140626
所有时间
961个事件
任务
完成
详细模式
事件(961)
统计信息
可视化
设定时间线的格式
缩小
放大
每列 1 秒
表格
格式
隐藏字段
所有字段
cellid
city
选定字段
_id
_index
_type
apn
bearer_json
cause
cellid
city
eNBip
eNBport
endDate
epsbearer_number
imei
imsi
interface
keyword
34605057 010
34605057 010
36703746 010
36703746 010
34672385 010
34672385 010
34672385 010
34672385 010
34664962 010
34664962 010
34664962 010
34664962 010
34664962 010
34664962 010
36703746 010
INDEX=lte_20140626 |TABLE endDate,mtmsi
饼图
格式
2014-06-26 10:08:32.040
2014-06-26 10:04:56.464
2014-06-26 10:08:32.031
2014-06-26 10:04:56.473
2014-06-26 10:06:35.276
2014-06-26 10:05:10.389
2014-06-26 10:06:35.276
2014-06-26 10:05:11.195
2014-06-26 10:06:35.276
2014-06-26 10:06:35.276
关于
支持
更新日志
文档
© 2008-2014 Eversec Inc. 保留所有权利。

感谢聆听!

- github
  - https://github.com/huangchen007/elasticsearch-rest-command
- weibo:
  - http://weibo.com/huang007
- qq群昵称/QQ号
  - C.Wong/3335956
- 姓名
  - 黄琛