[elasticsearch-查询基础篇](http://www.cnblogs.com/zhangchenliang/p/4195406.html)

elasticsearch的查询有两部分组成：query and filter。

两者的主要区别在于：filter是不计算相关性的，同时可以cache。因此，filter速度要快于query。

先记录一下es提供的各种query。

以下内容只为当做读书笔记，更多详细细节请参见<http://www.elasticsearch.org/guide/>

第一部分：query

在需要full-text-search和需要计算相关性的情况下，用query。而filter满足不了需求。

（1）match query and multi-match query //and match-all query and minimum should match query

match queries没有“query parsing”的过程，field不支持通配符，前缀等高级特性，只是参照指定的文本进行analysis，执行query，因此失败几率极小，适合search-box。

analyzed类型的query，故可指定analyzer

operator可指定or/and

zero-terms-query可指定none/all

cutoff-frequency可指定absolute值或者relative值

match-phase query可指定slot值，参见后续的search-in-depth

match-phase-prefix query可指定max\_expansion

（2）multi-match query

分别执行为单个field的match的查询。因此最终\_score值的计算规则各异。

fields可指定执行需要查询的字段，field可以支持通配符等高级特性（match query是不支持的），field可支持（^）指定各个field的boost权重

types可指定以下值，区分不同的查询行为：

best \_fields：\_score决定于得分最高的match-clause。field-centric

most\_fields：所有match-clause都会考虑在内。field-centric

cross-fields：把fileds当做一个big-fields。term-centric

phase and phase-prefix：每个field执行相应的query，combine the score

以上都有具体的应用场景和详细的计算规则，具体请参见后续的search-in-depth。

（3）bool query

一种复合查询，把其余类型的查询包裹进来。支持以下三种逻辑关系。

must： AND

must\_not：NOT

should：OR

（4）boosting query

一种复合查询，分为positive子查询和negitive子查询，两者的查询结构都会返回。

positive子查询的score保持不变，negetive子查询的值将会根据negative\_boost的值做相应程度的降低。

（5）common term query

一种略高级的查询，充分考虑了stop-word的低优先级，提高了查询精确性。

将terms分为了两种：more-importent（low-frequency） and less important（high-frequency）。less-important比如stop-words，eg：the and。

分组标准由cutoff\_frequence决定。两组query构成bool query。must应用于low\_frequence，should应用high\_frequence。

每一组内部都可以指定operator和mini\_should\_match。

如果group后只有一组，则默认退化为单组的子查询。

query执行中首先match到more-import这一组的doc，然后在这个基础上去match less-import，并且计算只计算match到的score。保证了效率，也充分考虑了relevance。

（6）constant score query

不计算相关性的query。沿用index过程中指定的score,。

（7）dismax query

对子查询的结果做union，score沿用子查询score的最大值。这种查询广泛应用于muti-field的查询。具体可以参见后续更新search-in-depth

（8）filtered query

combine another query with any fillter。

如果不指定query，默认为match\_all。当应用多个fitler的时候，可以指定strategy属性，expert-level。

（9）fuzzy query and fuzzy like this query and fuzzy like this field query

fuzzy query ：主要根据fuzziniess和prefix\_length进行匹配distance查询。根据type不同distance计算不一样。

numeric类型的distance类似于区间，string类型则依据Levenshtein distance，即从一个stringA变换到另一个stringB，需要变换的最小字母数。

如果指定为AUTO，则根据term的length有以下规则：

0-1：完全一致

1-4：1

>4：2

推荐指定prefix\_length，表明这个范围的字符需要精准匹配，如果不指定prefix\_lengh和fuzziniess参数，该查询负担较重。

（10）function score query

定义function去改变doc的score

（11）geoshape query

基于地理位置的查询

（12）has child query and has parent query and top children query

默认跟filter一样，query是包裹了一个constant\_score的filter。也有相关score的支持。

has\_child：匹配child字段，返回匹配到的对应的parent的结果。

has\_parent：匹配parent字段，返回匹配到对应child的结果。

top\_children query：has\_child query的一种，也是查询child字段，不过增加可控制参数，通过factor，incremental\_factor以及query的size来确定子查询的次数，直到满足

size为止，因此，可能需要多轮迭代子查询，所以total\_hits有可能是不准确的。

（13）ids query

查询指定id。

（14）indices query

在多个索引之中查询，允许提供一个indics参数指定将要查询的索引及相关的查询，同时指定no\_match\_query在indecs之外的索引中查询，返回结果。

（15）more like this and more like this field query

根据指定的like\_text，经过analysis生成若干个基于term的should查询合并成一个bool查询。

min\_term\_freq/max\_term\_freq/max\_term\_num：限制interesting term。

percentage\_terms\_to\_match：限制should查询应该满足的term比例。

more like this query 可指定多个field字段，more like this field query 则在一个field上查询。

（16）nested query

内嵌类型的查询，指定完整的path。

（17）prefix query

前缀查询。

（18）query string query and simple query string query

基于lucence查询语法的查询，指定字段/term/boost等。

simple query string query 跟 query string类似，这是会自动放弃invalid的部分，不会抛出异常。

默认的field是\_all。

（19）range query and regrex query and wildcard query

range query：区间查询，日期/string/num。

regrex query：正则查询。

wildcard query：通配符查询。

（20）span-\*query

（21）term query and terms query

基于term的查询。

（22）template query

注册一个查询模板，指定模板查询。

--------------------------

后续计划更新：

（1）一些特殊查询的比较。比如fuzzy 跟 more\_like等。

（2）search-in-depth

分类: [elasticsearch](http://www.cnblogs.com/zhangchenliang/category/643798.html" \t "_blank)