



MusesQL 用户手册

SUGON INFORMATION INDUSTRY CO.,LTD.



声明

本手册的用途在于帮助您正确地使用曙光公司大数据一体机产品(以下称"本产品"),在安装和第一次使用本产品前,请您务必先仔细阅读随机配送的所有资料,特别是本手册中所提及的注意事项。这会有助于您更好和安全地使用本产品。请妥善保管本手册,以便日后参阅。

本手册的描述并不代表对本产品规格和软硬件配置的任何说明。有关本产品的实际规格和配置,请查阅相关协议、装箱单、产品规格配置描述文件,或向产品的销售商咨询。

如您不正确地或未按本手册的指示和要求安装、使用或保管本产品,或让非曙光公司授权的技术人员修理、变更本产品,曙光公司将不对由此导致的损害承担任何责任。

本手册中所提供照片、图形、图表和插图,仅用于解释和说明目的,可能与实际产品有些差别,另外,产品实际规格和配置可能会根据需要不时变更,因此与本手册内容有所不同。请以实际产品为准。

本手册中所提及的非曙光公司网站信息,是为了方便起见而提供,此类网站中的信息不是曙光公司产品资料的一部分,也不是曙光公司服务的一部分,曙光公司对这些网站及信息的准确性和可用性不做任何保证。使用此类网站带来的风险将由您自行承担。

本手册不用于表明曙光公司对其产品和服务做了任何保证,无论是明示的还是默示的,包括(但不限于)本手册中推荐使用产品的适用性、安全性、适销性和适合某特定用途的保证。对本产品及相关服务的保证和保修承诺,应按可适用的协议或产品标准保修服务条款和条件执行。在法律法规的最大允许范围内,曙光公司对于您的使用或不能使用本产品而发生的任何损害(包括,但不限于直接或间接的个人损害、商业利润的损失、业务中断、商业信息的遗失或任何其他损失),不负任何赔偿责任。

对于您在本产品之外使用本产品随机提供的软件,或在本产品上使用非随机软件或经曙光公司认证推荐使用的专用软件之外的其他软件,曙光公司对其可靠性不做任何保证。

曙光公司已经对本手册进行了仔细的校勘和核对,但不能保证本手册完全没有任何错误和疏漏。为更好地



提供服务,曙光公司可能会对本手册中描述的产品软件和硬件及本手册的内容随时进行改进或更改,恕不另行通知。如果您在使用过程中发现本产品的实际情况与本手册有不一致之处,或您想得到最新的信息或有任何问题和想法,欢迎致电 400-810-0466 或登录曙光公司服务网站 www. sugon. com 垂询。

商标和版权

"SUGON"及图标是曙光信息产业股份有限公司的商标或注册商标。

"中科曙光"及图标是曙光信息产业股份有限公司的商标或注册商标,文中"曙光信息产业股份有限公司"简称"曙光公司"。

"Intel"、"Xeon"图标是 Intel 公司的注册商标。

"Microsoft"、"Windows"、"Windows Server"及"Windows Server System"是微软公司的商标或注册商标。

上面未列明的本手册提及的其他产品、标志和商标名称也可能是其他公司的商标或注册商标,并由其各自公司、其他性质的机构或个人拥有。

在本用户手册中描述的随机软件,是基于最终用户许可协议的条款和条件提供的,只能按照该最终用户许可协议的规定使用和复制。

版权所有©2013 曙光信息产业股份有限公司,所有权利保留。

本手册受到著作权法律法规保护,未经曙光信息产业股份有限公司事先书面授权,任何人士不得以任何方式对本手册的全部或任何部分进行复制、抄录、删减或将其编译为机读格式,以任何形式在可检索系统中存储,在有线或无线网络中传输,或以任何形式翻译为任何文字。





目录

声明2	2
商标和版权	3
目录	5
1 系统安装	õ
1.1 部署6	ō
1.2 参数配置6	õ
2 MQL 语法	3
2.1 语法形式8	3
2.2 管道	3
2.3 MQL 语句	3
2.3.1 search	3
2.3.2 aggregations	Э
2.3.3 sql	Э
2.4 过滤与聚合)
2.5 统计方法	2
2.6 过滤方法	5
3 查询服务17	7
3.1 WEB 服务17	7
3.2 用户权限	7
3.3 结果返回	7



1 系统安装

1.1 部署

准备工作:

Jdk8

Elasticsearch-1.7 或 Elasticsearch-2.x

Spark(1.5 对应 Elasticsearch-1.7,1.6 对应 Elasticsearch-2.x)

muses-analysis-service-xxx.jar

配置文件位于./configs/torchetl.properties 路径下,按照 1.2 节参数配置完毕后启动服务。启动服务命令:

java -cp muses-analysis-service-xxx:spark-assembly-xxx.jar com.sugon.muses.v1.Main --port 8080 (Elasticsearch-1.7)

java -cp muses-analysis-service-xxx:spark-assembly-xxx.jar com.sugon.muses.Main --port 8080 (Elasticsearch-2.x) 可通过 nohup 后台启动。

1.2 参数配置

参数	描述	示例
ESHost	多个用逗号分隔	127.0.0.1
ESPort		9300
ESPortForInsert		9200



ESClusterName		elasticsearch
ESPoolInitNum	Es 初始连接池大小	1
ESPoolMaxNum	Es 最大连接池大小	100
	单机(local)或分布	local
spark.master	式(spark://)	
es.mapping.id	配置 id 字段	_id
defeath to a	Es-1.7&Spark 需指定	
default.type	默认 type	type



2 MQL 语法

2.1 语法形式

MQL 语法采用管道方式,例如:

<MQL 语句> | <MQL 语句> | <MQL 语句> | ...

系统会默认依照管道顺序依次执行,并记录每条 MQL 语句返回的结果。

2.2 管道

MQL 中管道类似 Shell 中管道, Shell 中的管道是基于输出流的参数传递,而 MQL 中的管道用于基于结果集的参数传递,示例:

SEARCH table1 | cnt(1)

表示对 SEARCH table1 的结果集进一步求 count 数。

2.3 MQL **语句**

2.3.1 search

语法	SEARCH (<fields =="" csv=""> FROM)? <indexes =="" csv=""> (WHERE <filter>)? (OFFSET <offset =<="" th=""></offset></filter></indexes></fields>
	NUM>)? (LIMIT < limit = NUM>)? (ORDER BY < orderKey = STR> (ASC DESC)?)? (MAP
	<table_out =="" str="">)? (EXPORT <filename =="" str="">)? (INSERT INTO <indextypename =<="" td=""></indextypename></filename></table_out>
	STR>)?
	对指定索引进行检索:
	索引名为 CSV 格式的一个或多个索引,支持通配符"*",fields 为 csv 格式的字
描述	段名,可通过 WHERE 关键字指定过滤条件(详见 2.3 节),可通过 LIMIT 关键字
	限制返回记录条数,offset 指定查询偏移位置,可通过 MAP 关键字对结果集映射;
	EXPORT 和 INSERT INTO 可对返回结果导出到文件或索引
示例	SEARCH twitter



SEARCH user, age FROM twitter	B.(
SEARCH twitter LIMIT 10	
SEARCH twitter WHERE user = 'sugo	on' 15 and?
222	

2.3.2 aggregations

语法	(USING <table_in =="" str="">)? <aggregation></aggregation></table_in>
	(USING <table_in =="" str="">)? '(' <aggregation> ')' (', (' <aggregation> ')')+</aggregation></aggregation></table_in>
描述	执行聚合操作:
	执行聚合操作: 「USING < >)? ~~ 聚合 MQL 语句操作性能优于 SQL 操作,每条聚合 MQL 语句可以包含一个或多个
	聚合元素,且必须衔接 <search>的中间结果集,通过 USING 关键字可指定该 MQL</search>
	语句所衔接的中间结果集,具体聚合元素参见 2.3 节
示例	SEARCH twitter count(1) GROUP BY user
	SEARCH twitter MAP t1 USING t1 (count(1) GROUP BY user), (aggregation) ()

2.3.3 sql

语法	SQL'"' <sql =="" str="">'"' (MAP <table_out =="" str="">)? (EXPORT <filename =="" str="">)? (INSERT</filename></table_out></sql>
	INTO <indextypename =="" str="">)</indextypename>
	执行 SQL 查询操作:
描述	支持 SQL92 标准常用 SQL 语句,可通过 MAP 关键字对 SQL 查询的结果集映射;
	EXPORT 和 INSERT INTO 可对返回结果导出到文件或索引
示例	SQL "SELECT * FROM twitter WHERE user='sugon' "
	SEARCH twitter (sum(duration) AS s,count(1) AS c) GROUP BY user MAP tmp_table
	SQL "SELECT user, s/c FROM tmp_table" 引ん をか

2.4 过滤与聚合

filter

	<filter> OR <filter></filter></filter>
语法	<filter> AND <filter></filter></filter>
	NOT <filter></filter>



	'(' < filter > ')'
	<field =="" str=""> ('>' '<' = '>=' '<=' '!=') (' " '<str>' " ' <num>)</num></str></field>
	<func =="" function=""> To functions if this</func>
描述	执行过滤: 只是
	支持与、或、非逻辑运算(先级 AND 大于 OR)
	支持基于字段的比较运算
	支持返回值为布尔类型的过滤方法(详见 2.5 节)
示例	user = 'sugon' AND duration >=1000 AND duration <= 10000

aggregation

wheres (Werz'sugon' As a,

语法	<functions> (WHERE <filter>)? (WHERES '(' <filter> AS <wherename =="" str="">(, <filter></filter></wherename></filter></filter></functions>
	AS <wherename =="" str="">(+))')? (GROUP BY <groupings>)? (MAP <table_out =="" str="">)?</table_out></groupings></wherename>
	(EXPORT < fileName = STR>)? (INSERT INTO < indexT) peName = STR>)
描述	执行聚合方法: Regrouping, 是Stable
	可通过 WHERE/WHERES 关键字追加聚合的过滤条件及分组过滤条件,可通过
	GROUP BY 关键字指定聚合分组,可通过 MAP 关键字对聚合结果集映射,EXPORT
	和 INSERT INTO 可对返回结果导出到文件或索引
示例	count(1) GROUP BY user

functions

语法	<function> (''(' < function> (',' < function>)))</function>
描述	一个或多个方法。多个需使用小括号
示例	((sum(cost), count(1))

function

语法	<agg_function_name =="" str=""> '(' <csv> ')' (AS <column =="" str="">)?</column></csv></agg_function_name>
描述	agg_function_name 详见 2. 统计聚合方法



示例 count(1)

groupings

语法	<grouping></grouping>
	'[' <grouping> (',' <grouping> [+] ']'</grouping></grouping>
描述	一个或多个方法,多个需使用小括号
二個	user
示例	(user, post_date STEP 1m)

grouping

		<pre><field =="" str=""> STEP <step =="" num=""> (TO <end_field =="" str="">)? (HAVING COUNT</end_field></step></field></pre>	
		<min_count =="" num="">)?</min_count>	
		<field =="" str=""> STEP <step =="" str=""> (TO <end_field =="" str="">)? (HAVING COUNT <min_count< td=""><td></td></min_count<></end_field></step></field>	
	语法	= NUM>)?	
	市伝	<field =="" str=""> (HAVING COUNT <min_count =="" num="">)? (LIMIT limit = NUM>)? (ORDER</min_count></field>	こての 右
		BY <column =="" str=""> (ASC DESC)?)? 才製体心况是对某一以外的製 geo_distance(<field =="" str="">,<x =="" num="">,<y =="" num="">) STEP <step =="" str=""> TO <end =<="" td=""><td>1 20 7 TO 00000</td></end></step></y></x></field></column>	1 20 7 TO 00000
		geo_distance($<$ field = STR>, $<$ x = NUM>, $<$ y = NUM>) STEP $<$ step = STR> TO $<$ end =	TFWT I 20 OKNEK
		STR>?	
		基于数值字段的范围分组,发力1/2 2	
		基于时间字段的范围分组;	
		基于字符串型字段分组(支持排序);	
	描述	基于地域距离的分组。	
		其中:	
		end_field 可用于描述具有起始时间、结束时间的数据结构	
		min_count 指返回分组的最小支持度	
		cost STEP 10	① (
	示例	post_date STEP 1m TO finish_date HAVING COUNT 10	经面积库
不例	post_date STEP 1m TO finish_date HAVING COUNT 10 sum(cost) AS s user LIMIT 10 ORDER BY DESC 坦角与最为	12 (1)	
		geo_distance(location, 22, 116) STEP 100km TO 1000km	7mis

话。如此 11



2.5 统计方法 cnt (即 count) cnt(1) 语法 cnt(<field = STR>) 统计记录数 描述 统计指定字段层级下的记录数 count(1) 示例 SEARCH twitter Countill group by user count(user.*) sum wer time sum(<field = STR>) time user A 语法 sum(' " '<*script* = STR>' " ') sum(' " '<*script* = STR>' ", ' <*field* = STR>) 对指定字段求和 Count(1)=3 描述 使用脚本求和 Count (450x.+)=10 使用脚本求指定字段层级下的和 sum(duration) sum("doc['duration'].value / 60000") SUM [duratim] | 6000 [xhi]?

avg

示例

	avg(<field =="" str="">)</field>
语法	avg(' " ' <script =="" str="">' " ')</td></tr><tr><td>avg(' " '<script = STR>' ", ' <field = STR>)</td></tr><tr><td></td><td>对指定字段求平均值</td></tr><tr><td>描述</td><td>使用脚本求平均值</td></tr><tr><td></td><td>使用脚本求指定字段层级下的平均值</td></tr></tbody></table></script>

sum("doc['user.salary'].value / 30", user.*)



	avg(duration)
示例	avg("doc['duration'].value / 60000")
	avg("doc['user.salary'].value / 30", user.*)

max

	max(<field =="" str="">)</field>
语法	max(' " '< <i>script</i> = STR>' " ')
	max(' " '< <i>script</i> = STR>' ", ' < <i>field</i> = STR>)
	对指定字段求最大值
描述	使用脚本求最大值
	使用脚本求指定字段层级下的最大值
示例	max(duration)
	max("doc['duration'].value / 60000")
	max("doc['user.salary'].value / 30", user.*)

min

	min(<field =="" str="">)</field>
语法	min(' " '< <i>script</i> = STR>' " ')
	min(' " '< <i>script</i> = STR>' ", ' < <i>field</i> = STR>)
	对指定字段求最小值
描述	使用脚本求最小值
	使用脚本求指定字段层级下的最小值
	min(duration)
示例	min("doc['duration'].value / 60000")
	min("doc['user.salary'].value / 30", user.*)

distinct (即 distinct count)

	distinct(<field =="" str="">)</field>
语法	distinct(' " '< <i>script</i> = STR>' " ')



	distinct(' " ' <script =="" str="">' ", ' <field = STR>)</th></tr><tr><td></td><td>对指定字段求该值的种类数</td></tr><tr><td>描述</td><td>使用脚本求种类数</td></tr><tr><td></td><td>使用脚本求指定字段层级下的该值的种类数</td></tr><tr><td></td><td>distinct(duration)</td></tr><tr><td>示例</td><td>distinct("doc['duration'].value / 60000")</td></tr><tr><td></td><td>distinct("doc['user.salary'].value / 30", user.*)</td></tr></tbody></table></script>
--	--

top_hits

语法	top_hits(<field =="" str="">, <n =="" num="">)</n></field>
描述	返回每个分组的前 n 条记录,field 为排序字段
示例	top_hits(date, 5)

percentile

	percentile(<field =="" str="">)</field>	
语法	percentile(<field =="" str=""> (',' < percent = NUM>)+)</field>	
	percentile('"' <script =="" str="">!"') percent)(e('"'< Script = STR)'"'(</td><td>() < percent = MUM)+)</td></tr><tr><td>percentile(' " '<script = STR>' ", ' <field = STR>)</td><td>7</td></tr><tr><td></td><td>percentile(' " '<script = STR>' ", ' <field = STR> (',' <percent = NUM>)+)</td><td></td></tr><tr><td></td><td>采用默认百分比对指定字段求百分位数 でない。 うち合わまつ</td><td></td></tr><tr><td rowspan=4>描述</td><td>采用默认百分比对指定字段求百分位数 采用指定百分比对指定字段求百分位数 采用默认百分比使用脚本求百分位数 Percent — Jalua</td><td></td></tr><tr><td>采用默认百分比使用脚本求百分位数 Percent — Jalka</td><td></td></tr><tr><td>采用默认百分比使用脚本求指定字段层级下的百分位数</td><td></td></tr><tr><td>采用指定百分比使用脚本求指定字段层级下的百分位数</td><td></td></tr><tr><td></td><td>percentile(cost)</td><td></td></tr><tr><td rowspan=3>示例</td><td>返回: {"<u>1.0</u>":5.02,"<u>5.0</u>":5.1,"<u>25.0</u>":5.5,"<u>50.</u>0":6.0,"<u>75.0</u>":11.5,"<u>95.0</u>":15.9,"<u>99.0</u>":16.78}</td><td></td></tr><tr><td>percentile(cost, 25, 50, 75)</td><td></td></tr><tr><td>返回: {"25.0":5.5,"50.0":6.0,"75.0":11.5}</td><td></td></tr><tr><td></td><td>percentile("doc['duration'].value / 60000") ノッ, バ くっ</td><td></td></tr></tbody></table></script>	

14



percentile("doc['user.salary'].value / 30", user.*) percentile("doc['user.salary'].value / 30", user.*, 25, 50, 75)

percentile_ranks

油沙	percentile_ranks(<field =="" str=""> (',' <value= num="">)+)</value=></field>
语法	percentile_ranks(' " ' <script =="" str="">' ", ' <field = STR> (',' <value = NUM>)+)</td></tr><tr><td>描述</td><td>对指定字段的百分位数求百分比 百分位权 一> 百分化</td></tr><tr><td>抽处</td><td>使用脚本对百分位数求百分比 Value -> Dev Cent</td></tr><tr><th></th><th>percentile_ranks(cost, 5, 15, 25)</th></tr><tr><td rowspan=2>示例</td><td>返回:{"5.0":16.67,"1<mark>点</mark>0":28.79,"2<mark>点</mark>0":59.09}</td></tr><tr><td>percentile("doc['user.salary'].value / 30", user.*, 50, 100, 200)</td></tr></tbody></table></script>

geo_centroid

语法	geo_centroid(<field =="" str="">)</field>
描述	求指定地域字段的中心点
示例	geo_centroid(location)
71/101	返回:{"lat": 22.0, "lon": 116.0}

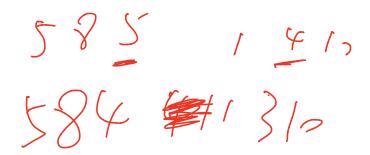
过滤方法 2.6

geo_bbox Ft. Ft.	
语法	geo_bbox(<field =="" str=""> ',' <top =="" num=""> ',' <left =="" num=""> ',' <box =="" num=""> ',' <right< th=""></right<></box></left></top></field>
	= NUM>)
描述	使用 bbox 对地域字段过滤
示例	SEARCH high_index WHERE geo_bbox(location, 22, 110, 18, 116)

geo_distance

geo_distance	
语法	geo_distance(<field =="" str=""> ',' <x =="" num=""> ',' <y =="" num=""> ',' <distance =="" str="">)</distance></y></x></field>
描述	过滤离指定地域点的距离大于指定距离的数据
示例	SEARCH high_index WHERE geo_distance(location, 20, 112, 100km)





16

曙光信息产业(北京)有限公司

www.sugon.com

5841 1510



3 查询服务

3.1 Web 服务

Web 服务采用 jetty 构造 WebService,提供 MusesQL 的查询服务, 含用户权限认证,Get 接口如下:

http://localhost:8080/muses?user=\${userName}&query=\${query} 其中:

\${userName}为用户,权限控制未开启时可省略 user; \${query}为查询的 MQL 语句。

3.2 用户权限

默认未开启,详见用户权限控制文档。

3.3 结果返回

返回结果为 Json 结构的 Map<String, List<Map<String, Object>>> 类型,其中,第一个 Map 的 key 为结果集名称,List 为每条记录(单 值返回则为只有 1 个元素的 List),List 中的 Map 为查询结果的字段-值。

示例(通过浏览器访问):

http://localhost:8080/muses?user=root&query=SEARCH twitter
MAP t|USING t (sum(duration) AS s,count(1) AS c) GROUP BY user MAP
tmp table|SQL "SELECT user, s/c FROM tmp table"

返回结果:

{"code":200,"_result":[{"user":"sugon","_c1":11.5}","{"user":"k","_c



1":5.0}]}