



dreamstime.com

Banco de Dados não Relacionais – NoSQL
Prof.: Henrique Batista da Silva

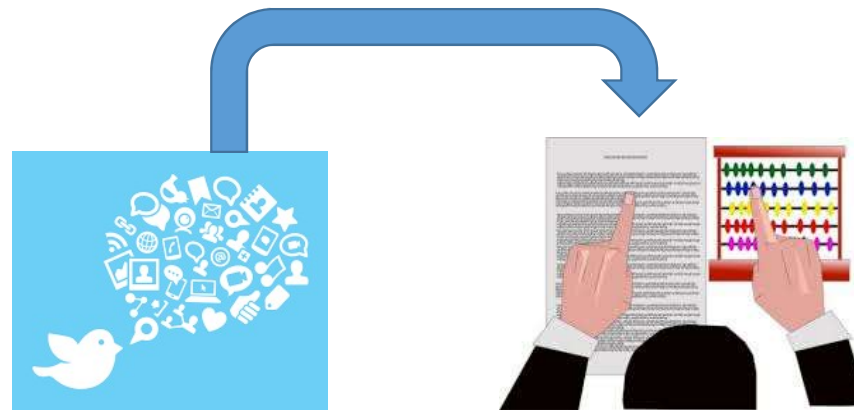
Trabalho Prático: ***Contador de Palavras***

Cristiane Ferreira Tavares Carvalhaes
Lourdes Souza
Nelson Nunes
Vanesca Freitas Dias

- ☐ Tema escolhido;
- ☐ Metodologia aplicada;
- ☐ Algoritmos e resultados;
- ☐ Uso do Banco de Dados Não Relacional : Escolha e Motivadores.

Tema escolhido

Como tema para este trabalho, foi optado pelo desenvolvimento de uma solução de contagem de palavras através de textos coletados do Twitter.

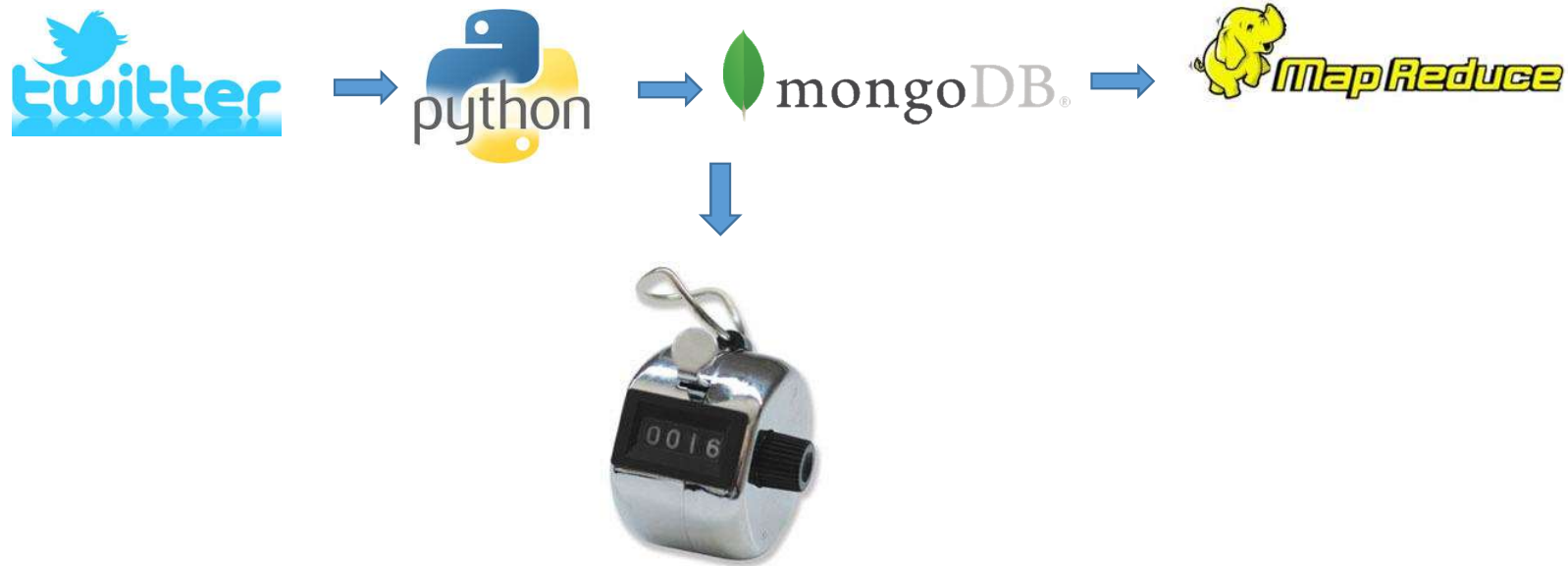


Metodologia Aplicada – Contador Palavras

A metodologia aplicada para o desenvolvimento desta solução, utilizou-se de dois recursos:

1. Desenvolvimento de algoritmo em Python com o uso do API Twitter : Tweepy, para coleta dos textos do Twitter e armazenamento no banco de dados não relacional MongoDB;
2. Aplicação de algoritmo MapReduce para contagem de palavras no MongoDB;

Metodologia Aplicada – Contador Palavras

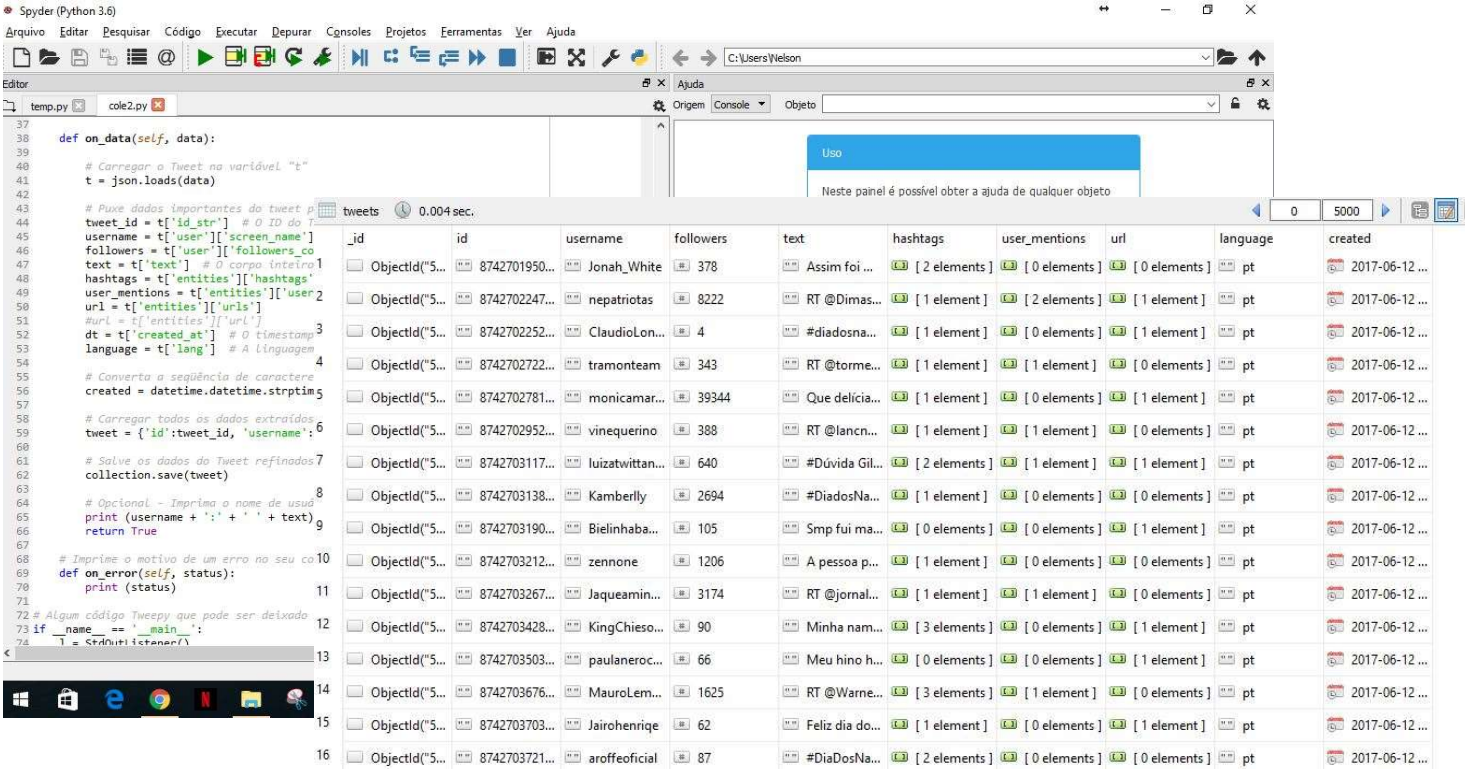


Algoritmos e resultados

Passo 1: Coleta twitters

Desenvolvimento de algoritmo em Python com o uso das APIs:

- PyMongo
- Tweepy



The screenshot shows the Spyder Python IDE with a script named `temp.py` and a variable explorer showing a list of tweets. The script defines a function `on_data(self, data)` that processes tweet data and saves it to a MongoDB collection. The variable explorer shows a list of tweets with columns: `_id`, `id`, `username`, `followers`, `text`, `hashtags`, `user_mentions`, `url`, `language`, and `created`.

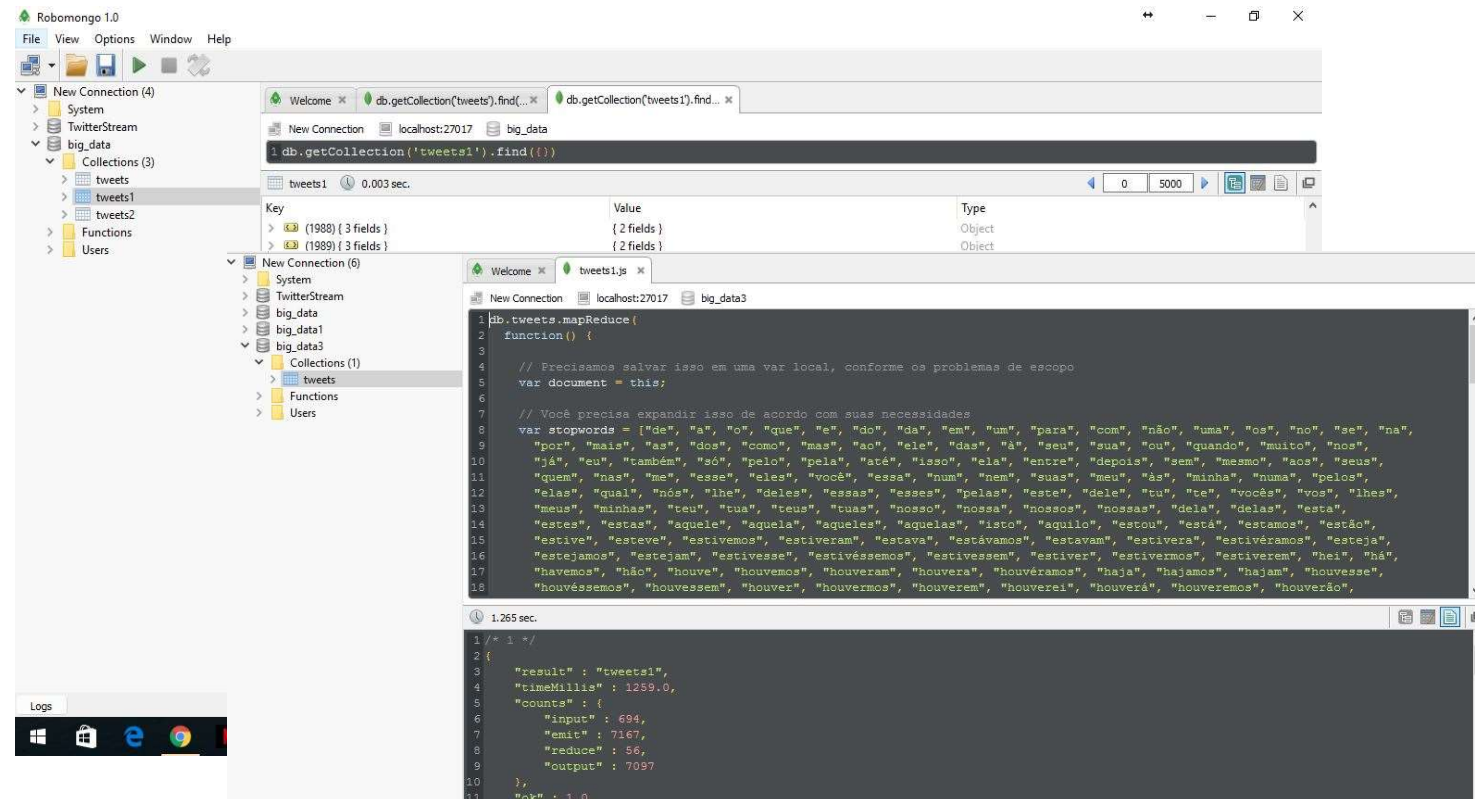
_id	id	username	followers	text	hashtags	user_mentions	url	language	created
Objectid("5...	8742701950...	Jonah_White	378	Assim foi ...	[2 elements]	[0 elements]	[0 elements]	pt	2017-06-12 ...
Objectid("5...	8742702247...	nepatriotas	8222	RT @Dimas...	[1 element]	[2 elements]	[1 element]	pt	2017-06-12 ...
Objectid("5...	8742702252...	ClaudioLon...	4	#diadosna...	[1 element]	[0 elements]	[1 element]	pt	2017-06-12 ...
Objectid("5...	8742702722...	tramonteam	343	RT @torme...	[1 element]	[1 element]	[0 elements]	pt	2017-06-12 ...
Objectid("5...	8742702781...	monicamar...	39344	Que delicia...	[1 element]	[0 elements]	[1 element]	pt	2017-06-12 ...
Objectid("5...	8742702952...	vinequerino	388	RT @lancn...	[1 element]	[1 element]	[0 elements]	pt	2017-06-12 ...
Objectid("5...	8742703117...	luizattwittan...	640	#Dúvida Gil...	[2 elements]	[1 element]	[1 element]	pt	2017-06-12 ...
Objectid("5...	8742703138...	Kamberlly	2694	#DiadosNa...	[1 element]	[0 elements]	[0 elements]	pt	2017-06-12 ...
Objectid("5...	8742703190...	Bielinhaba...	105	Smp fui ma...	[0 elements]	[0 elements]	[1 element]	pt	2017-06-12 ...
Objectid("5...	8742703212...	zennone	1206	A pessoa p...	[1 element]	[0 elements]	[0 elements]	pt	2017-06-12 ...
Objectid("5...	8742703267...	Jaqueamin...	3174	RT @jornal...	[1 element]	[1 element]	[0 elements]	pt	2017-06-12 ...
Objectid("5...	8742703428...	KingChieso...	90	Minha nam...	[3 elements]	[0 elements]	[1 element]	pt	2017-06-12 ...
Objectid("5...	8742703503...	paulanero...	66	Meu hino h...	[0 elements]	[0 elements]	[1 element]	pt	2017-06-12 ...
Objectid("5...	8742703676...	MauroLem...	1625	RT @Warne...	[3 elements]	[1 element]	[0 elements]	pt	2017-06-12 ...
Objectid("5...	8742703703...	Jairohenriq...	62	Feliz dia do...	[1 element]	[0 elements]	[1 element]	pt	2017-06-12 ...
Objectid("5...	8742703721...	aroffeocial	87	#DiaDosNa...	[2 elements]	[0 elements]	[0 elements]	pt	2017-06-12 ...

Algoritmos e resultados

Passo 2: Aplicação stopwords e limpeza base

Execução de algoritmo para:

1. Aplicação das stopwords e limpeza da base;



The screenshot shows the Robomongo 1.0 interface. On the left, a tree view shows the database structure with collections 'tweets1', 'tweets2', and 'Users'. The main window displays a query result for 'tweets1' with 2 fields. Below this, a new connection (6) shows a mapReduce operation being executed on the 'tweets' collection. The mapReduce function is defined as follows:

```

1 db.tweets.mapReduce(
2   function() {
3
4     // Precisamos salvar isso em uma var local, conforme os problemas de escopo
5     var document = this;
6
7     // Você precisa expandir isso de acordo com suas necessidades
8     var stopwords = ["de", "a", "o", "que", "e", "do", "da", "em", "um", "para", "com", "não", "uma", "os", "no", "se", "na",
9       "por", "mais", "as", "dos", "como", "mas", "ao", "ele", "das", "à", "seu", "sua", "ou", "quando", "muito", "nos",
10      "já", "eu", "também", "só", "pelo", "pela", "até", "isso", "ela", "entre", "depois", "sem", "mesmo", "aos", "seus",
11      "quem", "nas", "me", "esse", "eles", "você", "essa", "num", "nem", "suas", "meu", "às", "minha", "numa", "pelos",
12      "elas", "qual", "nós", "lhe", "deles", "essas", "esses", "pelas", "este", "dele", "tu", "te", "você", "você", "lhes",
13      "meus", "minhas", "teu", "tua", "teus", "tuas", "nosso", "nossa", "nossos", "nossas", "dela", "delas", "esta",
14      "estes", "estas", "aquele", "aquela", "aqueles", "aquelas", "isto", "aquilo", "estou", "está", "estamos", "estão",
15      "estive", "esteve", "estivemos", "estiveram", "estava", "estávamos", "estavam", "estiverá", "estiveremos", "estêja",
16      "estejamos", "estejam", "estivesse", "estivessemos", "estivessem", "estiver", "estivermos", "estiverem", "hei", "há",
17      "havemos", "hão", "houve", "houvemos", "houveram", "houvera", "houvéramos", "haja", "hajamos", "hajam", "houvesse",
18      "houvéssemos", "houvessem", "houver", "houvermos", "houverem", "houverei", "houverá", "houveremos", "houverão",
19    ],
20    // ... (rest of the function code)
21  },
22  {
23    // ... (mapReduce options)
24  }
25);


```

The result of the mapReduce operation is shown in the bottom window, indicating that the operation was successful and the output was saved to the 'tweets1' collection.

Algoritmos e resultados

Passo 3: MapReduce e agrupamento palavras

Execução de algoritmo
em JavaScript para
aplicação de
MapReduce e
agrupamento de
palavras.



New Connection localhost:27017 big_data3

```
1 db.tweets1.aggregate(  
2   // Nós combinamos insensível a maiúsculas e minúsculas ("i") como queremos impedir  
3   // erros de digitação para reduzir nossos resultados de pesquisa  
4   {$match: {"_id.word": /^S/i }},  
5   {$group: {  
6     // Aqui  
7     // cria Key  
8     _id: "$_id.word",  
9     occurrences: {  
10      // ...  
11      $push: "$_id.word",  
12      doc: "$_id.doc",  
13      field: "$_id.field",  
14      score: "$_id.score",  
15    }  
16  }  
17  // Note  
18  // para
```

tweets2 0.003 sec.

Key	Value	Type
_id: "quer"	{ 3 fields }	Object
occurrences	{ 3 fields }	Object
_id: "quer"	{ 3 fields }	Object
doc	{ 3 fields }	Object
field	{ 3 fields }	Object
score	{ 3 fields }	Object
occurrences	{ 3 fields }	Object
_id	quer	String
occurrences	[11 elements]	Array
[0]	{ 2 fields }	Object
doc	ObjectId("593ea4b066ac9415f06deac4")	ObjectId
field	text	String
score	11.0	Double
occurrences	{ 2 fields }	Object
[1]	{ 2 fields }	Object
doc	ObjectId("593ea4dd66ac9415f06dead2")	ObjectId
field	text	String
score	11.0	Double
occurrences	{ 2 fields }	Object
[2]	{ 2 fields }	Object
[3]	{ 2 fields }	Object
[4]	{ 2 fields }	Object
[5]	{ 2 fields }	Object
[6]	{ 2 fields }	Object
[7]	{ 2 fields }	Object
[8]	{ 2 fields }	Object
[9]	{ 2 fields }	Object
[10]	{ 2 fields }	Object
score	11.0	Double
occurrences	{ 3 fields }	Object
[259]	{ 3 fields }	Object
[260]	{ 3 fields }	Object

0.078 sec.

```
25 /* 3 */  
26 {  
27   "id": "  
28   "occurrences": {  
29     {  
30     {  
31     {  
32     {  
33     {  
34     {  
35     {  
36     {  
37     {  
38     {  
39     {  
40     {  
41     {  
42     {  
43     {  
44     {  
45     {  
46     {  
47     {  
48     {  
49     {  
50     {  
51     {  
52     {  
53     {  
54     {  
55     {  
56     {  
57     {  
58     {  
59     {  
60     {  
61     {  
62     {  
63     {  
64     {  
65     {  
66     {  
67     {  
68     {  
69     {  
70     {  
71     {  
72     {  
73     {  
74     {  
75     {  
76     {  
77     {  
78     {  
79     {  
80     {  
81     {  
82     {  
83     {  
84     {  
85     {  
86     {  
87     {  
88     {  
89     {  
90     {  
91     {  
92     {  
93     {  
94     {  
95     {  
96     {  
97     {  
98     {  
99     {  
100    {  
101    {  
102    {  
103    {  
104    {  
105    {  
106    {  
107    {  
108    {  
109    {  
110    {  
111    {  
112    {  
113    {  
114    {  
115    {  
116    {  
117    {  
118    {  
119    {  
120    {  
121    {  
122    {  
123    {  
124    {  
125    {  
126    {  
127    {  
128    {  
129    {  
130    {  
131    {  
132    {  
133    {  
134    {  
135    {  
136    {  
137    {  
138    {  
139    {  
140    {  
141    {  
142    {  
143    {  
144    {  
145    {  
146    {  
147    {  
148    {  
149    {  
150    {  
151    {  
152    {  
153    {  
154    {  
155    {  
156    {  
157    {  
158    {  
159    {  
160    {  
161    {  
162    {  
163    {  
164    {  
165    {  
166    {  
167    {  
168    {  
169    {  
170    {  
171    {  
172    {  
173    {  
174    {  
175    {  
176    {  
177    {  
178    {  
179    {  
180    {  
181    {  
182    {  
183    {  
184    {  
185    {  
186    {  
187    {  
188    {  
189    {  
190    {  
191    {  
192    {  
193    {  
194    {  
195    {  
196    {  
197    {  
198    {  
199    {  
200    {  
201    {  
202    {  
203    {  
204    {  
205    {  
206    {  
207    {  
208    {  
209    {  
210    {  
211    {  
212    {  
213    {  
214    {  
215    {  
216    {  
217    {  
218    {  
219    {  
220    {  
221    {  
222    {  
223    {  
224    {  
225    {  
226    {  
227    {  
228    {  
229    {  
230    {  
231    {  
232    {  
233    {  
234    {  
235    {  
236    {  
237    {  
238    {  
239    {  
240    {  
241    {  
242    {  
243    {  
244    {  
245    {  
246    {  
247    {  
248    {  
249    {  
250    {  
251    {  
252    {  
253    {  
254    {  
255    {  
256    {  
257    {  
258    {  
259    {  
260    {  
261    {  
262    {  
263    {  
264    {  
265    {  
266    {  
267    {  
268    {  
269    {  
270    {  
271    {  
272    {  
273    {  
274    {  
275    {  
276    {  
277    {  
278    {  
279    {  
280    {  
281    {  
282    {  
283    {  
284    {  
285    {  
286    {  
287    {  
288    {  
289    {  
290    {  
291    {  
292    {  
293    {  
294    {  
295    {  
296    {  
297    {  
298    {  
299    {  
300    {  
301    {  
302    {  
303    {  
304    {  
305    {  
306    {  
307    {  
308    {  
309    {  
310    {  
311    {  
312    {  
313    {  
314    {  
315    {  
316    {  
317    {  
318    {  
319    {  
320    {  
321    {  
322    {  
323    {  
324    {  
325    {  
326    {  
327    {  
328    {  
329    {  
330    {  
331    {  
332    {  
333    {  
334    {  
335    {  
336    {  
337    {  
338    {  
339    {  
340    {  
341    {  
342    {  
343    {  
344    {  
345    {  
346    {  
347    {  
348    {  
349    {  
350    {  
351    {  
352    {  
353    {  
354    {  
355    {  
356    {  
357    {  
358    {  
359    {  
360    {  
361    {  
362    {  
363    {  
364    {  
365    {  
366    {  
367    {  
368    {  
369    {  
370    {  
371    {  
372    {  
373    {  
374    {  
375    {  
376    {  
377    {  
378    {  
379    {  
380    {  
381    {  
382    {  
383    {  
384    {<
```


Algoritmos e resultados

Resultados

New Connection localhost:27017 big_data3

```

1 db.tweets1.aggregate(
2 {
3   $group:{
4     _id:"$_id.word",
5     occurrences:{ $push:{doc:"$_id.doc",
6       score:{$sum:"$value"}
7     }
8   },{
9     $out:"tweets2"
10  }
11 )

```

0.502 sec.

1 Fetched 0 record(s) in 0ms

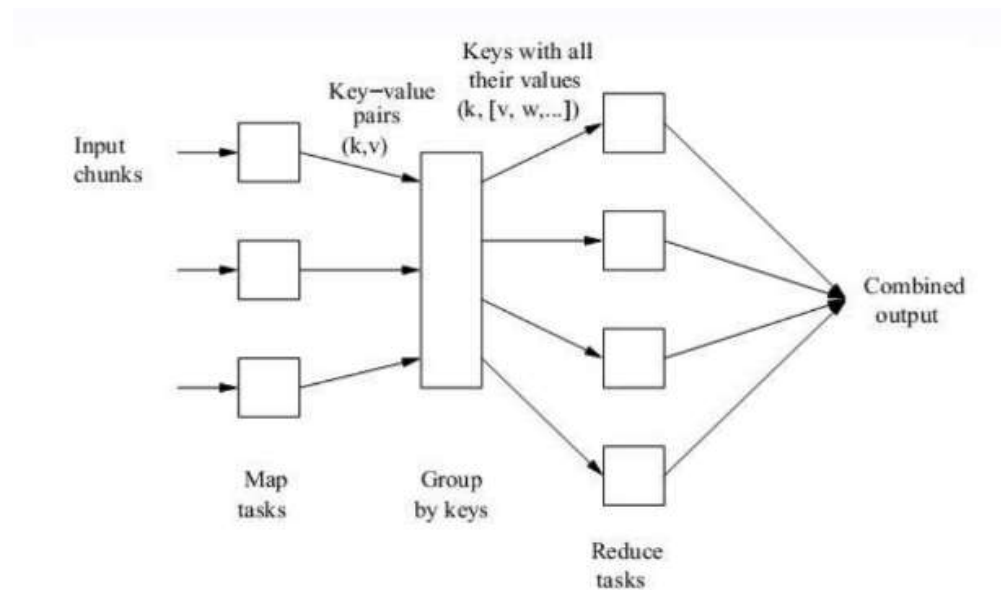
2

tweets2 0.003 sec.		
_id	occurrences	score
tal	[1 element]	1.0
rap10	[1 element]	1.0
qur	[1 element]	1.0
quis	[1 element]	1.0
que...	[1 element]	1.0
questionou	[1 element]	1.0
queridos	[1 element]	1.0
queria	[1 element]	1.0
quer	[11 elements]	11.0
quente	[1 element]	1.0
que..❤️#Pl...	[2 elements]	2.0
qno	[1 element]	1.0
puder	[1 element]	1.0
prova.."	[1 element]	1.0
proporcionar	[1 element]	1.0
promoção	[4 elements]	4.0

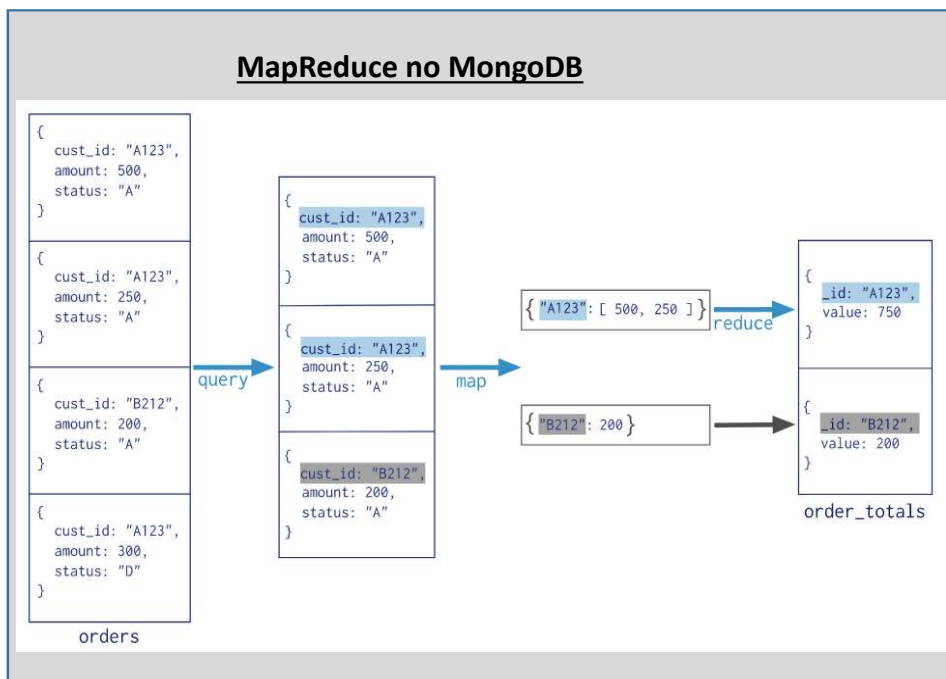
MapReduce

Processamento de dados para condensação de grandes volumes em resultados agregados. Utilizada a operação de redução de mapa, através do comando MapReduce disponível no MongoDB.

Esquema de funcionamento do MapReduce



MapReduce - MongoDB



Redução do mapa, aplicado pelo MongoDB, a cada documento de entrada (documentos na coleção) com:

- Emissão pares de valores-chave. Para as chaves que possuem vários valores, o MongoDB aplica a fase de redução, que coleta e condensa os dados agregados;
- Armazenamento dos resultados em uma coleção;
- Execução de todas as funções de redução de mapa em JavaScript com execução dentro do processo mongod;

Banco de Dados Não Relacional - MongoDB

A escolha do uso do banco de dados não relacional MongoDB, se deve:

- Possibilidade de manipulação de alto volume de dados não estruturado;
- Existência de função MapReduce que permitisse realizar o trabalho proposto de contagem de palavras, através do processo de agregação de palavras.

Algoritmos utilizados



Coleta Twitters



MapReduce MongoDB

