

# 使用redis bitmap来计算开机数

需求：

统计全国所有机器的开机数据

思路：

恰好redis提供了bitmap数据结构，可以利用其来统计

假设全国有10000台机器，时间粒度为秒。redis所占用的大小= $10000 \times 60 \times 60 \times 24$  大概10.41kbyte,每天归档后清理key即可

也可以用在：签到、计算活跃用户

实践：

```
1 # -*- coding: utf-8 -*-
2 import time
3 import redis
4
5 redis_bit_map = {
6     "\x00": [],
7     "\x01": [7],
8     "\x02": [6],
9     "\x03": [6, 7],
10    "\x04": [5],
11    "\x05": [5, 7],
12    "\x06": [5, 6],
13    "\x07": [5, 6, 7],
```

```
14  "\x08": [4],
15  "\t": [4, 7],
16  "
17  ": [4, 6],
18  "\x0b": [4, 6, 7],
19  "\x0c": [4, 5],
20  "\r": [4, 5, 7],
21  "\x0e": [4, 5, 6],
22  "\x0f": [4, 5, 6, 7],
23  "\x10": [3],
24  "\x11": [3, 7],
25  "\x12": [3, 6],
26  "\x13": [3, 6, 7],
27  "\x14": [3, 5],
28  "\x15": [3, 5, 7],
29  "\x16": [3, 5, 6],
30  "\x17": [3, 5, 6, 7],
31  "\x18": [3, 4],
32  "\x19": [3, 4, 7],
33  "\x1a": [3, 4, 6],
34  "\x1b": [3, 4, 6, 7],
35  "\x1c": [3, 4, 5],
36  "\x1d": [3, 4, 5, 7],
37  "\x1e": [3, 4, 5, 6],
38  "\x1f": [3, 4, 5, 6, 7],
39  " ": [2],
40  "!": [2, 7],
41  "\"": [2, 6],
42  "#": [2, 6, 7],
43  "$": [2, 5],
44  "%": [2, 5, 7],
45  "&": [2, 5, 6],
46  "'": [2, 5, 6, 7],
47  "(" : [2, 4],
48  ")" : [2, 4, 7],
49  "*" : [2, 4, 6],
50  "+" : [2, 4, 6, 7],
51  "," : [2, 4, 5],
52  "-" : [2, 4, 5, 7],
53  "." : [2, 4, 5, 6],
54  "/" : [2, 4, 5, 6, 7],
55  "0" : [2, 3],
56  "1" : [2, 3, 7],
57  "2" : [2, 3, 6],
58  "3" : [2, 3, 6, 7],
59  "4" : [2, 3, 5],
```

```
60 "5": [2, 3, 5, 7],
61 "6": [2, 3, 5, 6],
62 "7": [2, 3, 5, 6, 7],
63 "8": [2, 3, 4],
64 "9": [2, 3, 4, 7],
65 " ": [2, 3, 4, 6],
66 ";": [2, 3, 4, 6, 7],
67 "<": [2, 3, 4, 5],
68 "=": [2, 3, 4, 5, 7],
69 ">": [2, 3, 4, 5, 6],
70 "?": [2, 3, 4, 5, 6, 7],
71 "@": [1],
72 "A": [1, 7],
73 "B": [1, 6],
74 "C": [1, 6, 7],
75 "D": [1, 5],
76 "E": [1, 5, 7],
77 "F": [1, 5, 6],
78 "G": [1, 5, 6, 7],
79 "H": [1, 4],
80 "I": [1, 4, 7],
81 "J": [1, 4, 6],
82 "K": [1, 4, 6, 7],
83 "L": [1, 4, 5],
84 "M": [1, 4, 5, 7],
85 "N": [1, 4, 5, 6],
86 "O": [1, 4, 5, 6, 7],
87 "P": [1, 3],
88 "Q": [1, 3, 7],
89 "R": [1, 3, 6],
90 "S": [1, 3, 6, 7],
91 "T": [1, 3, 5],
92 "U": [1, 3, 5, 7],
93 "V": [1, 3, 5, 6],
94 "W": [1, 3, 5, 6, 7],
95 "X": [1, 3, 4],
96 "Y": [1, 3, 4, 7],
97 "Z": [1, 3, 4, 6],
98 "[": [1, 3, 4, 6, 7],
99 "\\": [1, 3, 4, 5],
100 "]" : [1, 3, 4, 5, 7],
101 "^": [1, 3, 4, 5, 6],
102 "_": [1, 3, 4, 5, 6, 7],
103 "`": [1, 2],
104 "a": [1, 2, 7],
105 "b": [1, 2, 6],
```

```
106 "c": [1, 2, 6, 7],
107 "d": [1, 2, 5],
108 "e": [1, 2, 5, 7],
109 "f": [1, 2, 5, 6],
110 "g": [1, 2, 5, 6, 7],
111 "h": [1, 2, 4],
112 "i": [1, 2, 4, 7],
113 "j": [1, 2, 4, 6],
114 "k": [1, 2, 4, 6, 7],
115 "l": [1, 2, 4, 5],
116 "m": [1, 2, 4, 5, 7],
117 "n": [1, 2, 4, 5, 6],
118 "o": [1, 2, 4, 5, 6, 7],
119 "p": [1, 2, 3],
120 "q": [1, 2, 3, 7],
121 "r": [1, 2, 3, 6],
122 "s": [1, 2, 3, 6, 7],
123 "t": [1, 2, 3, 5],
124 "u": [1, 2, 3, 5, 7],
125 "v": [1, 2, 3, 5, 6],
126 "w": [1, 2, 3, 5, 6, 7],
127 "x": [1, 2, 3, 4],
128 "y": [1, 2, 3, 4, 7],
129 "z": [1, 2, 3, 4, 6],
130 "{": [1, 2, 3, 4, 6, 7],
131 "|": [1, 2, 3, 4, 5],
132 "}": [1, 2, 3, 4, 5, 7],
133 "~": [1, 2, 3, 4, 5, 6],
134 "\x7f": [1, 2, 3, 4, 5, 6, 7],
135 "\x80": [0],
136 "\x81": [0, 7],
137 "\x82": [0, 6],
138 "\x83": [0, 6, 7],
139 "\x84": [0, 5],
140 "\x85": [0, 5, 7],
141 "\x86": [0, 5, 6],
142 "\x87": [0, 5, 6, 7],
143 "\x88": [0, 4],
144 "\x89": [0, 4, 7],
145 "\x8a": [0, 4, 6],
146 "\x8b": [0, 4, 6, 7],
147 "\x8c": [0, 4, 5],
148 "\x8d": [0, 4, 5, 7],
149 "\x8e": [0, 4, 5, 6],
150 "\x8f": [0, 4, 5, 6, 7],
151 "\x90": [0, 3],
```

152 "\\x91": [0, 3, 7],  
153 "\\x92": [0, 3, 6],  
154 "\\x93": [0, 3, 6, 7],  
155 "\\x94": [0, 3, 5],  
156 "\\x95": [0, 3, 5, 7],  
157 "\\x96": [0, 3, 5, 6],  
158 "\\x97": [0, 3, 5, 6, 7],  
159 "\\x98": [0, 3, 4],  
160 "\\x99": [0, 3, 4, 7],  
161 "\\x9a": [0, 3, 4, 6],  
162 "\\x9b": [0, 3, 4, 6, 7],  
163 "\\x9c": [0, 3, 4, 5],  
164 "\\x9d": [0, 3, 4, 5, 7],  
165 "\\x9e": [0, 3, 4, 5, 6],  
166 "\\x9f": [0, 3, 4, 5, 6, 7],  
167 "\\xa0": [0, 2],  
168 "\\xa1": [0, 2, 7],  
169 "\\xa2": [0, 2, 6],  
170 "\\xa3": [0, 2, 6, 7],  
171 "\\xa4": [0, 2, 5],  
172 "\\xa5": [0, 2, 5, 7],  
173 "\\xa6": [0, 2, 5, 6],  
174 "\\xa7": [0, 2, 5, 6, 7],  
175 "\\xa8": [0, 2, 4],  
176 "\\xa9": [0, 2, 4, 7],  
177 "\\xaa": [0, 2, 4, 6],  
178 "\\xab": [0, 2, 4, 6, 7],  
179 "\\xac": [0, 2, 4, 5],  
180 "\\xad": [0, 2, 4, 5, 7],  
181 "\\xae": [0, 2, 4, 5, 6],  
182 "\\xaf": [0, 2, 4, 5, 6, 7],  
183 "\\xb0": [0, 2, 3],  
184 "\\xb1": [0, 2, 3, 7],  
185 "\\xb2": [0, 2, 3, 6],  
186 "\\xb3": [0, 2, 3, 6, 7],  
187 "\\xb4": [0, 2, 3, 5],  
188 "\\xb5": [0, 2, 3, 5, 7],  
189 "\\xb6": [0, 2, 3, 5, 6],  
190 "\\xb7": [0, 2, 3, 5, 6, 7],  
191 "\\xb8": [0, 2, 3, 4],  
192 "\\xb9": [0, 2, 3, 4, 7],  
193 "\\xba": [0, 2, 3, 4, 6],  
194 "\\xbb": [0, 2, 3, 4, 6, 7],  
195 "\\xbc": [0, 2, 3, 4, 5],  
196 "\\xbd": [0, 2, 3, 4, 5, 7],  
197 "\\xbe": [0, 2, 3, 4, 5, 6],

```
198  "\xbf": [0, 2, 3, 4, 5, 6, 7],
199  "\xc0": [0, 1],
200  "\xc1": [0, 1, 7],
201  "\xc2": [0, 1, 6],
202  "\xc3": [0, 1, 6, 7],
203  "\xc4": [0, 1, 5],
204  "\xc5": [0, 1, 5, 7],
205  "\xc6": [0, 1, 5, 6],
206  "\xc7": [0, 1, 5, 6, 7],
207  "\xc8": [0, 1, 4],
208  "\xc9": [0, 1, 4, 7],
209  "\xca": [0, 1, 4, 6],
210  "\xcb": [0, 1, 4, 6, 7],
211  "\xcc": [0, 1, 4, 5],
212  "\xcd": [0, 1, 4, 5, 7],
213  "\xce": [0, 1, 4, 5, 6],
214  "\xcf": [0, 1, 4, 5, 6, 7],
215  "\xd0": [0, 1, 3],
216  "\xd1": [0, 1, 3, 7],
217  "\xd2": [0, 1, 3, 6],
218  "\xd3": [0, 1, 3, 6, 7],
219  "\xd4": [0, 1, 3, 5],
220  "\xd5": [0, 1, 3, 5, 7],
221  "\xd6": [0, 1, 3, 5, 6],
222  "\xd7": [0, 1, 3, 5, 6, 7],
223  "\xd8": [0, 1, 3, 4],
224  "\xd9": [0, 1, 3, 4, 7],
225  "\xda": [0, 1, 3, 4, 6],
226  "\xdb": [0, 1, 3, 4, 6, 7],
227  "\xdc": [0, 1, 3, 4, 5],
228  "\xdd": [0, 1, 3, 4, 5, 7],
229  "\xde": [0, 1, 3, 4, 5, 6],
230  "\xdf": [0, 1, 3, 4, 5, 6, 7],
231  "\xe0": [0, 1, 2],
232  "\xe1": [0, 1, 2, 7],
233  "\xe2": [0, 1, 2, 6],
234  "\xe3": [0, 1, 2, 6, 7],
235  "\xe4": [0, 1, 2, 5],
236  "\xe5": [0, 1, 2, 5, 7],
237  "\xe6": [0, 1, 2, 5, 6],
238  "\xe7": [0, 1, 2, 5, 6, 7],
239  "\xe8": [0, 1, 2, 4],
240  "\xe9": [0, 1, 2, 4, 7],
241  "\xea": [0, 1, 2, 4, 6],
242  "\xeb": [0, 1, 2, 4, 6, 7],
243  "\xec": [0, 1, 2, 4, 5],
```

```

244 "\xed": [0, 1, 2, 4, 5, 7],
245 "\xee": [0, 1, 2, 4, 5, 6],
246 "\xef": [0, 1, 2, 4, 5, 6, 7],
247 "\xf0": [0, 1, 2, 3],
248 "\xf1": [0, 1, 2, 3, 7],
249 "\xf2": [0, 1, 2, 3, 6],
250 "\xf3": [0, 1, 2, 3, 6, 7],
251 "\xf4": [0, 1, 2, 3, 5],
252 "\xf5": [0, 1, 2, 3, 5, 7],
253 "\xf6": [0, 1, 2, 3, 5, 6],
254 "\xf7": [0, 1, 2, 3, 5, 6, 7],
255 "\xf8": [0, 1, 2, 3, 4],
256 "\xf9": [0, 1, 2, 3, 4, 7],
257 "\xfa": [0, 1, 2, 3, 4, 6],
258 "\xfb": [0, 1, 2, 3, 4, 6, 7],
259 "\xfc": [0, 1, 2, 3, 4, 5],
260 "\xfd": [0, 1, 2, 3, 4, 5, 7],
261 "\xfe": [0, 1, 2, 3, 4, 5, 6],
262 "\xff": [0, 1, 2, 3, 4, 5, 6, 7],
263 }
264
265
266 def find_bit(bits):
267     """
268     返回这个区间中该位对应的偏移量
269     >>> find_bit(b"\x80")
270     [0]
271     >>> find_bit("@")
272     [1]
273     >>> find_bit(b"\xff\xff")
274     [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
275     >>> find_bit(b"\x11")
276     [3, 7]
277     >>> find_bit(b"\x01\x01")
278     [7, 15]
279     >>> find_bit(b"\x11\x11")
280     [3, 7, 11, 15]
281     """
282     return (no * 8 + j for no, i in enumerate(bits) for j in redis_bit_map[chr(i)])
283
284
285 def git_bits(rc, key, fr, to=10):
286     """ 按位查询 返回bit位为1的下标
287     :param fr: 开始位置
288     :param to: 结束位置
289     :return: []

```

```
290     """
291     # end 必须按8取整
292     (st, offset), ed, l = divmod(fr, 8), (fr + to + 7) // 8, []
293     for _id in find_bit(rc.getrange(key, st, ed)):
294         # 偏移量大出要查询的范围直接终止
295         if _id > offset + to:
296             break
297         # 偏移量小于起始位置
298         if _id < offset:
299             continue
300         # 区间外的偏移量(st * 8) + 区间内的偏移量
301         l.append(st * 8 + _id)
302     return l
303
304
305
306 if __name__ == "__main__":
307     # redis key a value: bitmap
308     # 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 84400 84500 84600
309     # 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 1
310     r = redis.Redis('10.9.36.222')
311     print(git_bits(r, 'a', 0, 10)) # [2, 4, 8, 9]
312     print(git_bits(r, 'a', 0, 84600)) # [2, 4, 8, 9, 14, 15, 84400, 84500, 84600]
```