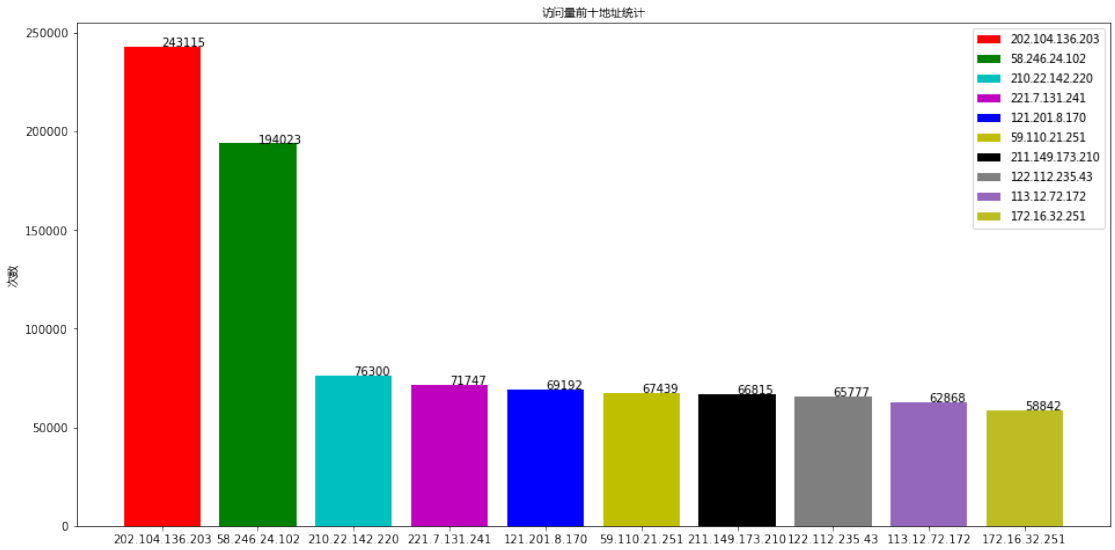
Untitled3

August 31, 2017

```
In [1]: import pyspark

In [2]: sc=pyspark.SparkContext('local[*]')

In [3]: compress_logs = sc.wholeTextFiles("hdfs://172.19.6.59:9000/nginx")

In [4]: import re

In [5]: onlyAccessLogs=compress_logs.filter(lambda a: re.search('access', a[0]))

In [6]: splitLogs=onlyAccessLogs.flatMap(lambda a: a[1].split('\n'))

In [7]: def getIP(str):
            ip=re.findall(r'^\d+\.\d+\.\d+\.\d+', str)
            return ip

In [8]: accessIP=splitLogs.map(getIP) #IP

In [9]: strIP=accessIP.map(lambda a: ''.join(a)) #

In [63]: strIP.count()

Out[63]: 1665310

In [10]: singleIP=strIP.distinct() #

In [13]: singleIP.count()

Out[13]: 2532

In [11]: fullIP=strIP.distinct().collect() #ip

In [12]: from operator import add

In [13]: frequencyIP=strIP.map(lambda a: (a, 1)).reduceByKey(add)

In [14]: topTenIP=frequencyIP.sortBy(lambda a: a[1], ascending=False).take(10)
```

In [15]:
```
import matplotlib.pyplot as plt
ips=topTenIP
import matplotlib.font_manager as fm
myfont = fm.FontProperties(fname='/opt/conda/pkgs/matplotlib-2.0.2-py36_2/lib/python3.6
hight=(ips[0][1],ips[1][1],ips[2][1],ips[3][1],ips[4][1],ips[5][1],ips[6][1],ips[7][1],
left=(0,1,2,3,4,5,6,7,8,9)
ip=(ips[0][0],ips[1][0],ips[2][0],ips[3][0],ips[4][0],ips[5][0],ips[6][0],ips[7][0],ips
#plt.xlabel("IP")
color=("r","g","c","m","b","y","k","tab:gray","tab:purple","tab:olive",)
plt.figure(figsize=(16,8))
#XY
plt.ylabel("", fontproperties=myfont)
#
plt.xticks(left, ip)
#
plt.title("", fontproperties=myfont)
#
rect = plt.bar(left=left, height=hight, align="center", color=color)
#
chuid=rect.get_children()
#legend
plt.legend(chuid, ip, prop=myfont)
# plt.legend((rect,),("list",))
#
for i in range(10):
    plt.text(left[i],hight[i],hight[i])
plt.show()
```



In [17]:
```
splitLogs.take(5)
```

```
Out[17]: ['172.16.40.16 - - [25/Aug/2017:03:43:02 +0800] "GET /home HTTP/1.1" 200 4309 "-" "Zabb
          '211.149.173.210 - - [25/Aug/2017:03:43:03 +0800] "GET /sockjs/info HTTP/1.1" 200 90 "
          '210.22.142.220 - - [25/Aug/2017:03:43:03 +0800] "GET /_timesync HTTP/1.1" 200 13 "htt
          '172.16.32.251 - - [25/Aug/2017:03:43:05 +0800] "GET /sockjs/info HTTP/1.1" 200 90 "-"
          '121.201.8.170 - - [25/Aug/2017:03:43:06 +0800] "GET /home HTTP/1.1" 200 1762 "-" "pyt

In [76]: #
         def getStatus(str):
             #ip=re.findall(r'^\d+\.\d+\.\d+\.\d+', str)
             logList=str.split()
             if len(logList) > 9:
                 statusNum=logList[8]
                 return statusNum

In [77]: accessStatus = splitLogs.map(getStatus)

In [81]: reduceStatus=accessStatus.map(lambda a: (a, 1)).reduceByKey(add)

In [83]: sortStatus=reduceStatus.sortBy(lambda a: a[1], ascending=False)

In [111]: def checkStatus(s):
              if s[0] == None:
                  return False
              elif s[0] == '"-"':
                  return False
              else:
                  return True

In [114]: filterStatus=sortStatus.filter(checkStatus).collect()

In [116]: print(filterStatus)

[('200', 1232613), ('304', 285013), ('204', 64180), ('404', 40327), ('101', 26826), ('206', 9537


In [119]: #
          plt.figure(figsize=(12,8))
          ips=filterStatus #
          #
          labels=[ips[0][0],ips[1][0],ips[2][0],ips[3][0],ips[4][0],ips[5][0],ips[6][0],ips[7][0
          # print(labels)
          sizes = [ips[0][1],ips[1][1],ips[2][1],ips[3][1],ips[4][1],ips[5][1],ips[6][1],ips[7][
          # print(sizes)
          colors = ['red','yellowgreen','lightskyblue','yellow','green','black','white','orange'
          explode = (0.01,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
          patches,l_text,p_text = plt.pie(sizes,explode=explode,labels=ips,colors=colors,
                                          labeldistance = 1,autopct = '%3.1f%%',shadow = False,
                                          startangle = 90,pctdistance = 0.6)
          for t in l_text:
```

```
    t.set_size=(3)
for t in p_text:
    t.set_size=(3)
plt.title('Access status')
plt.axis('equal')
plt.legend()
plt.show()
```