

# 现代程序设计第11周作业

---

谢奕飞 20377077

## ReadData函数

```
1 def ReadData(file_path):
2     '''
3     读取json文件,返回行的列表
4     '''
5     result=[]
6     with open(file_path,encoding='utf8') as f:
7         data=json.load(f)
8         for line in tqdm(data,desc='reading...'):
9             result.append(line.get('content'))
10    return result
```

## LoadStopwords函数

```
1 def LoadStopwords(dict_path):
2     '''
3     jieba载入分词库,返回分词集合
4     '''
5     jieba.load_userdict(dict_path)
6     f=open(dict_path,'r',encoding='utf8')
7     stop_words={line.strip() for line in f.readlines()}
8     f.close()
9     return stop_words
```

## Map函数

```
1 def Map(data,stop_words,result):
2     '''
3     Map进程读取文档并进行词频统计,返回该文本的词频统计结果
4     '''
5     count_dict={}
6     for line in data:
7         words=jieba.lcut(line)
8         for word in words:
9             if word in stop_words:
10                continue
11             elif word not in count_dict:
12                 count_dict.update({word:0})
13             count_dict[word]+=1
```

```
14 result.append(count_dict)
```

## Reduce函数

```
1 def Reduce(result_lis,save_path):
2     '''
3     整合所有结果
4     '''
5     result_all={}
6     for result in tqdm(result_lis,desc='reducing...'):
7         for key,value in result.items():
8             if key not in result_all:
9                 result_all.update({key:0})
10            result_all[key]+=value
11 with open('test.csv','w',newline='',encoding='utf8') as f:
12     writer=csv.writer(f)
13     for row in result_all.items():
14         writer.writerow(row)
```

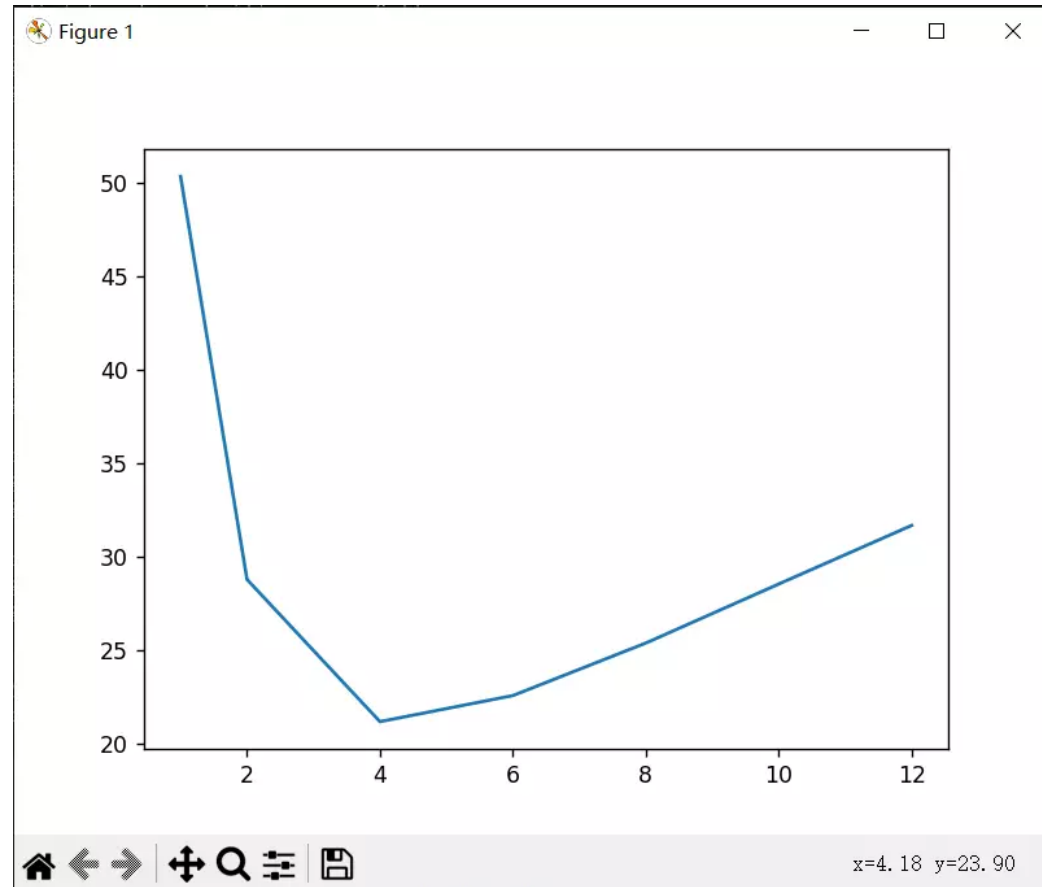
## main函数

```
1 if __name__=='__main__':
2     file_path='D:/Project/Python/week11MapReduce/sohu_data.json'
3     dict_path='D:/Project/Python/week11MapReduce/stopwords_list.txt'
4     save_path='D:/Project/Python/week11MapReduce/result.csv'
5     #N=[psutil.cpu_count(False)]
6     #N=[1,2,3,4,5,6,7,8,9,10,11,12]
7     N=[1,2,4,6,8,10,12]
8     data=ReadData(file_path)
9     stop_words=LoadStopwords(dict_path)
10    N_time=[]
11    #data=data[:10000] #减小数据量
12    size=len(data)
13    for n in N:
14        p_list=[]
15        m=Manager()
16        result=m.list([])
17        for i in range(n):#创建CPU内核数个进程
18            p=Process(target=Map,args=(data[int(size/n*i):int(size/n*(i+1))])
19            p_list.append(p)
20        start_time=time.time()
21        for p in p_list:
22            p.start() #启动进程
23        for p in p_list:
24            p.join() #阻滞主进程
25        t=time.time()-start_time
26        print('共用时: {}s'.format(t)) #测试总用时
27        N_time.append(t)
28        print('{}进程处理完成'.format(n))
29    Reduce(result,save_path)
```

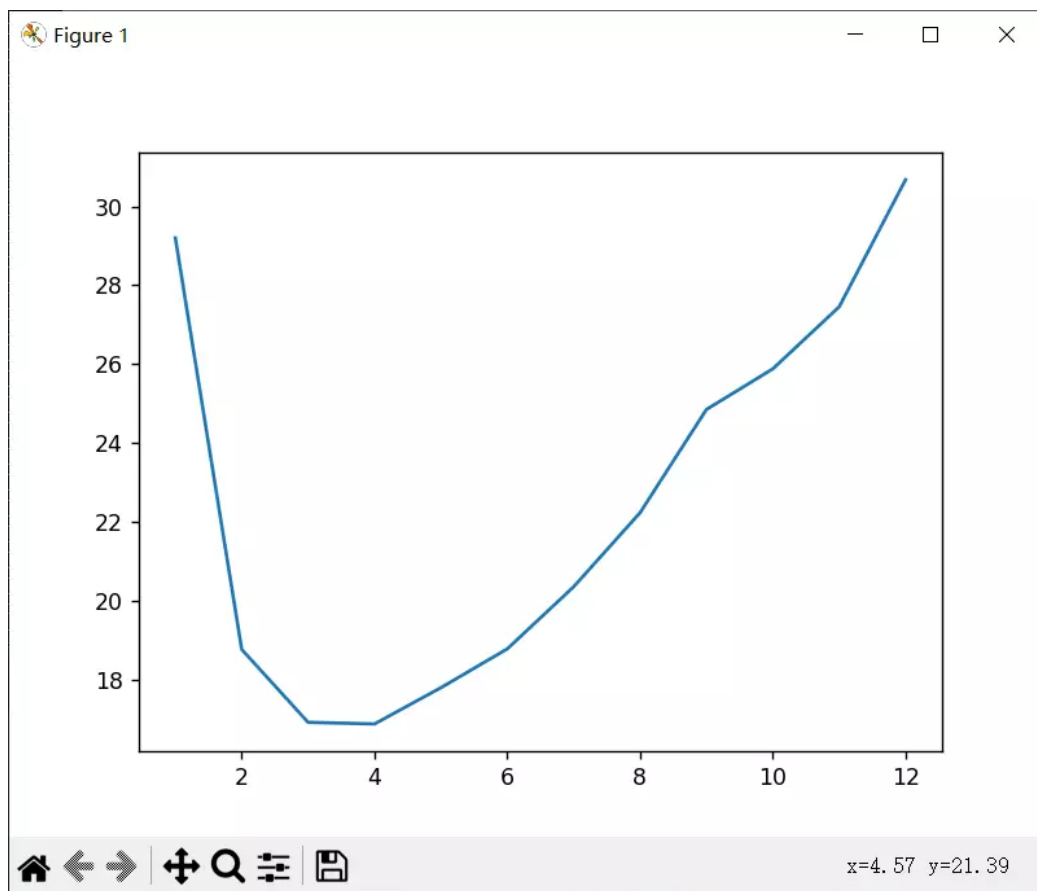
```
30 plt.plot(N,N_time)
31 plt.show()
```

## 运行结果

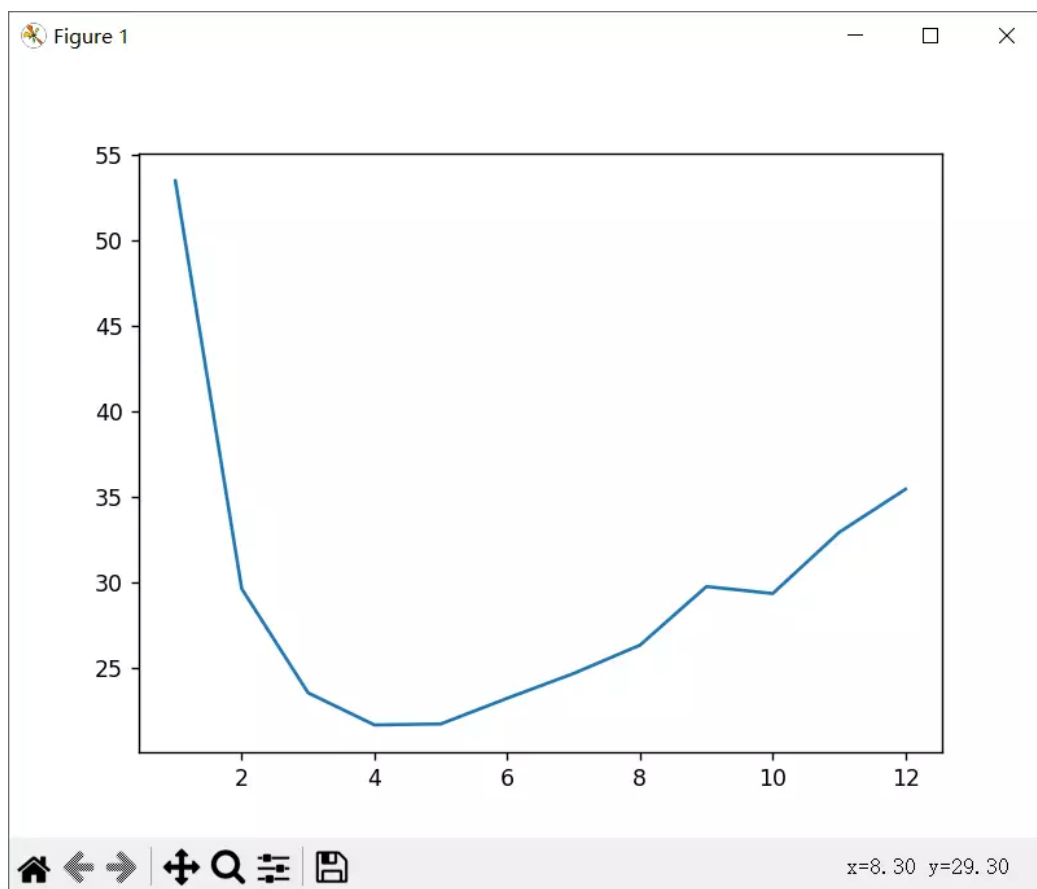
```
1 N=[1,2,4,6,8,10,12]
2 data=data[:10000]#减小数据量
```



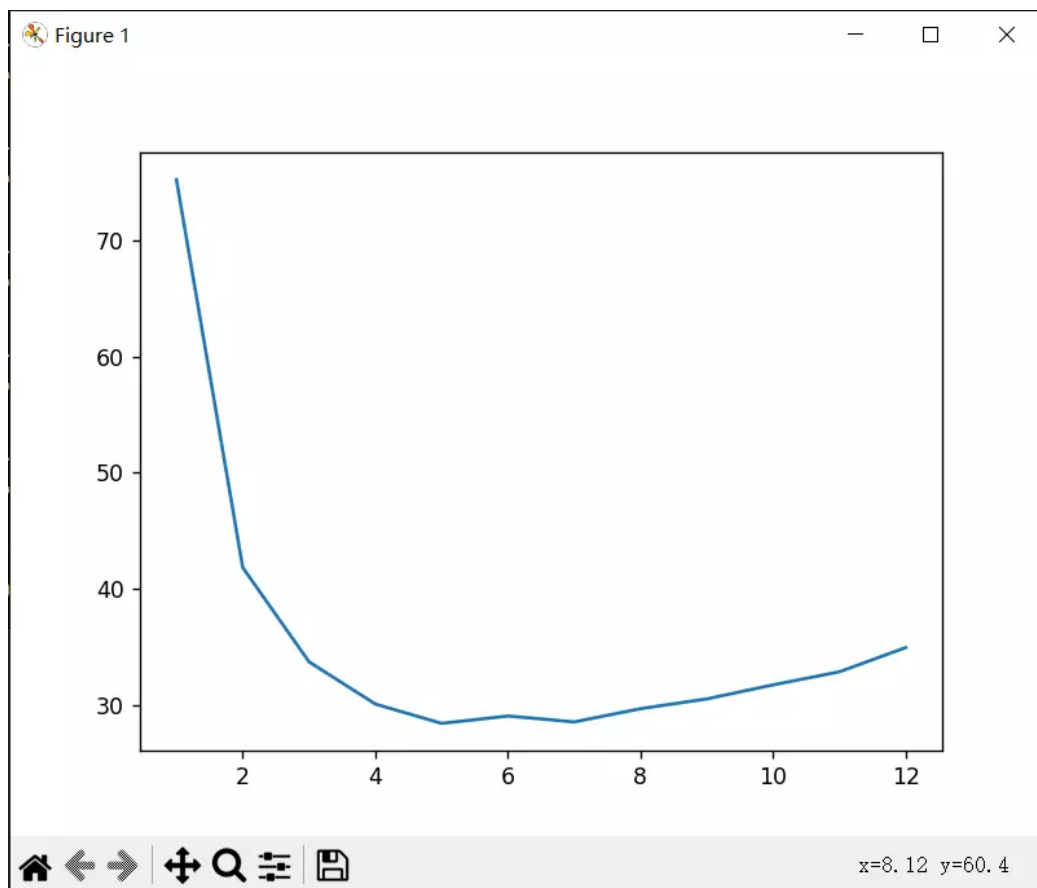
```
1 N=[1,2,3,4,5,6,7,8,9,10,11,12]
2 data=data[:5000]#减小数据量
```



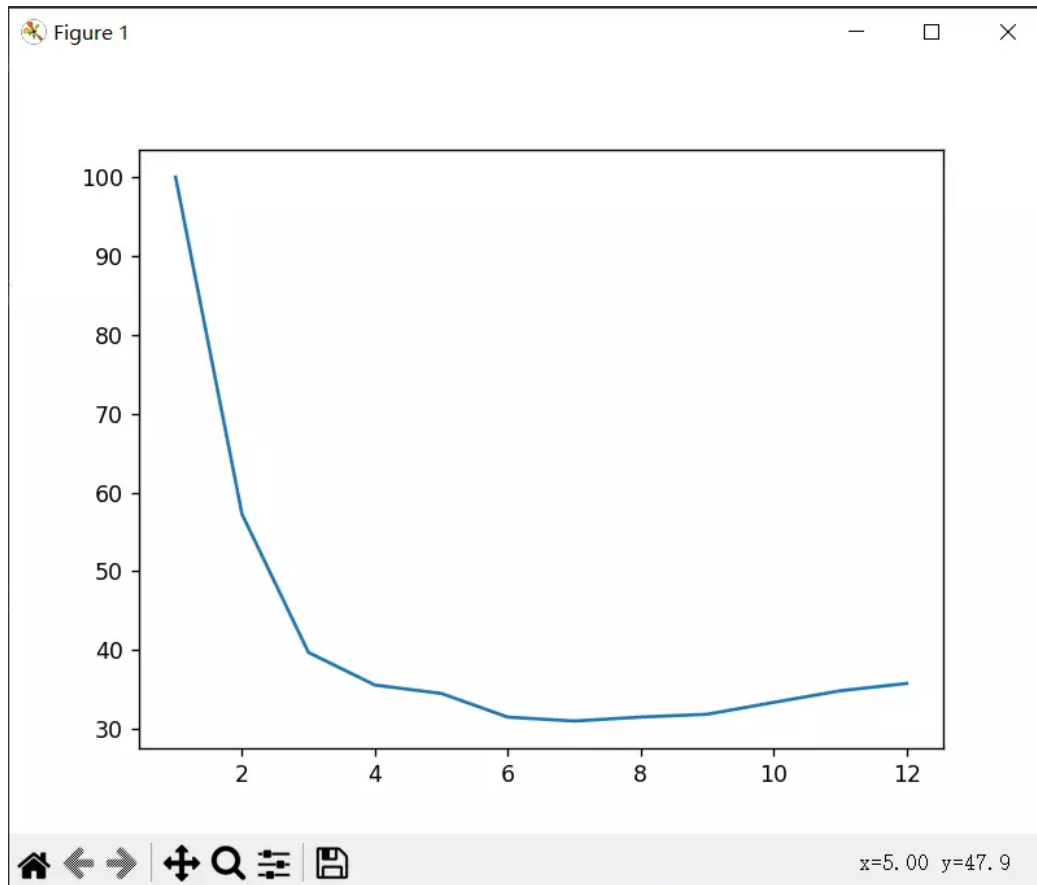
```
1 N=[1,2,3,4,5,6,7,8,9,10,11,12]
2 data=data[:10000]#减小数据量
```



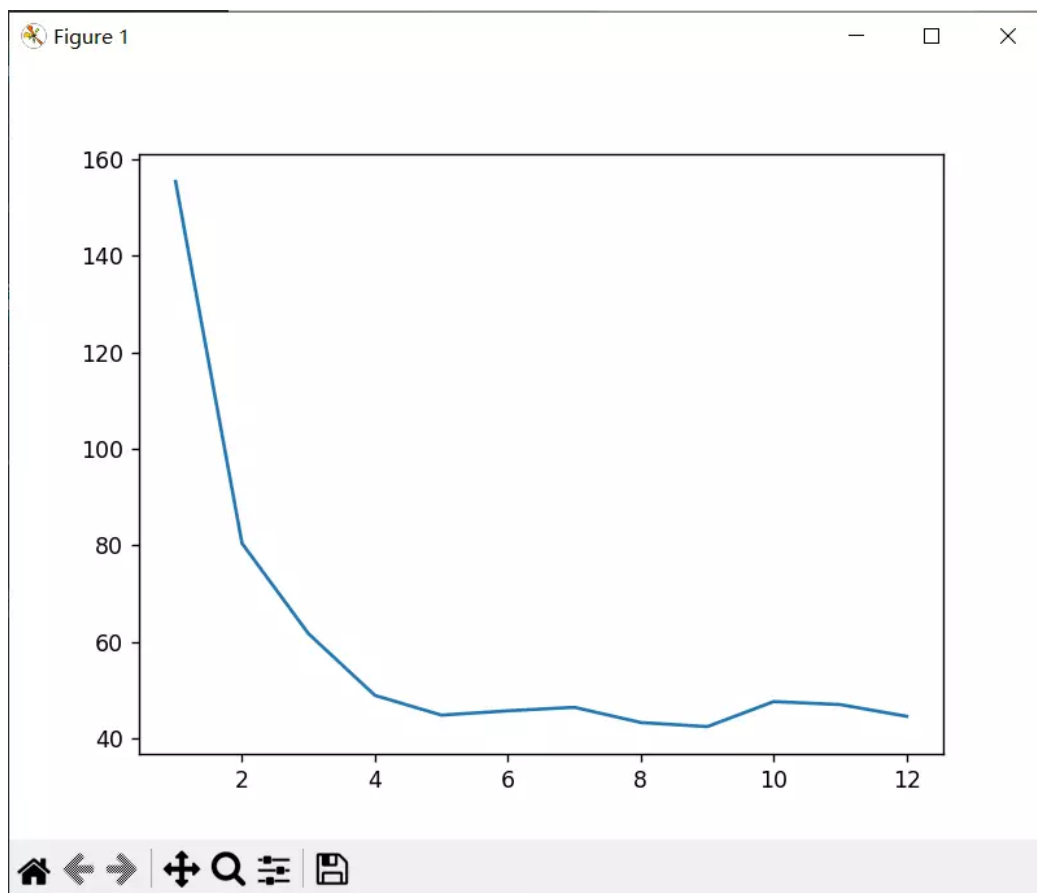
```
1 N=[1,2,3,4,5,6,7,8,9,10,11,12]
2 data=data[:15000]#减小数据量
```



```
1 N=[1,2,3,4,5,6,7,8,9,10,11,12]
2 data=data[:20000]#减小数据量
```

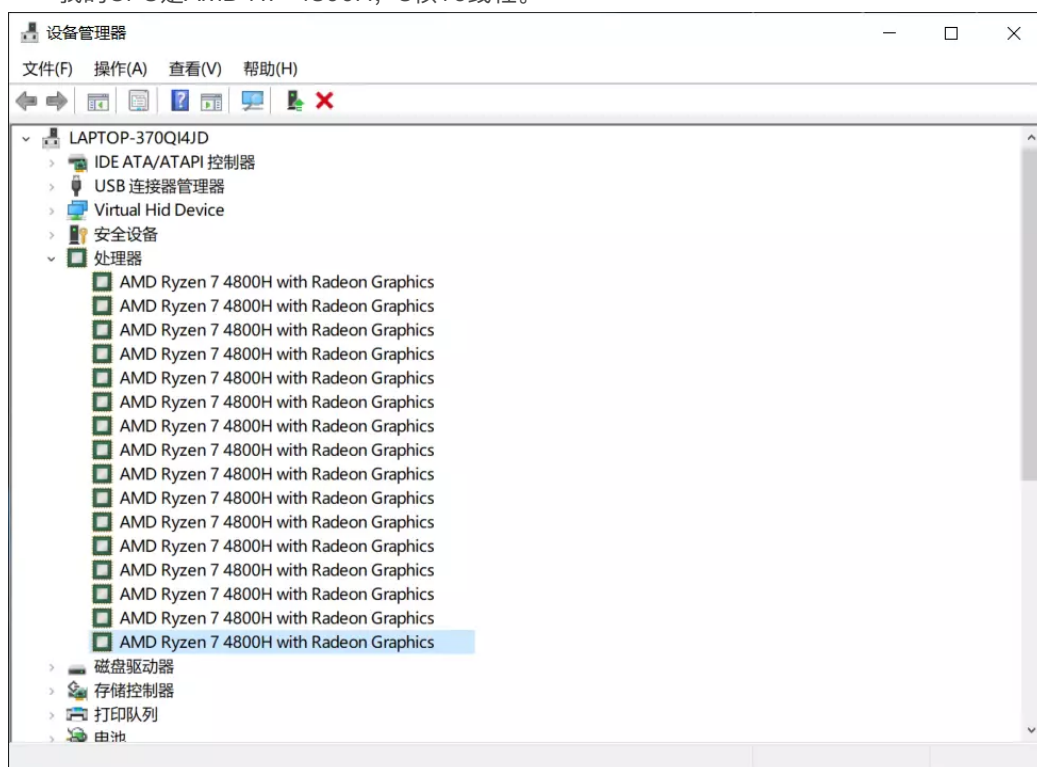


```
1 N=[1,2,3,4,5,6,7,8,9,10,11,12]
2 data=data[:30000]#减小数据量
```



## 分析

我的CPU是AMD R7-4800H，8核16线程。



数据量	最短用时的进程数
5000	4
10000	4
15000	5
20000	7
30000	9

当数据量比较小时（10000行）最短用时的进程数为4或5；随着数据量增大（30000行），最短用时的进程数逐渐接近电脑核数8。

## 输出

```

d:\Project\Python - VS Code 控制台
Loading model cost 0.603 seconds.
Prefix dict has been built successfully.
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.626 seconds.
Prefix dict has been built successfully.
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.615 seconds.
Prefix dict has been built successfully.
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.645 seconds.
Prefix dict has been built successfully.
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.627 seconds.
Prefix dict has been built successfully.
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.602 seconds.
Prefix dict has been built successfully.
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.582 seconds.
Prefix dict has been built successfully.
共用时: 31.68583583831787s
12进程处理完成
reducing...: 100% | 12/12 [00:00<00:00, 35.05it/s]
```

## 详细输出

以

```

1 N=[1,2,4,6,8,10,12]
2 data=data[:10000]#减小数据量
```

为例

```

1 reading...: 100% | 1245835/1245835 [00:01<00:00, 693798.41it/s]
2 Building prefix dict from the default dictionary ...
3 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
4 Loading model cost 0.553 seconds.
5 Prefix dict has been built successfully.
6 Building prefix dict from the default dictionary ...
7 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
8 Loading model cost 0.564 seconds.
9 Prefix dict has been built successfully.
10 共用时: 50.357043504714966s
11 1进程处理完成
12 Building prefix dict from the default dictionary ...
13 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
14 Loading model cost 0.572 seconds.
```

```
15 Prefix dict has been built successfully.
16 Building prefix dict from the default dictionary ...
17 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
18 Loading model cost 0.567 seconds.
19 Prefix dict has been built successfully.
20 共用时: 28.794530391693115s
21 2进程处理完成
22 Building prefix dict from the default dictionary ...
23 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
24 Loading model cost 0.586 seconds.
25 Prefix dict has been built successfully.
26 Building prefix dict from the default dictionary ...
27 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
28 Loading model cost 0.585 seconds.
29 Prefix dict has been built successfully.
30 Building prefix dict from the default dictionary ...
31 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
32 Loading model cost 0.621 seconds.
33 Prefix dict has been built successfully.
34 Building prefix dict from the default dictionary ...
35 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
36 Loading model cost 0.620 seconds.
37 Prefix dict has been built successfully.
38 共用时: 21.189520120620728s
39 4进程处理完成
40 Building prefix dict from the default dictionary ...
41 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
42 Loading model cost 0.593 seconds.
43 Prefix dict has been built successfully.
44 Building prefix dict from the default dictionary ...
45 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
46 Loading model cost 0.590 seconds.
47 Prefix dict has been built successfully.
48 Building prefix dict from the default dictionary ...
49 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
50 Loading model cost 0.604 seconds.
51 Prefix dict has been built successfully.
52 Building prefix dict from the default dictionary ...
53 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
54 Loading model cost 0.611 seconds.
55 Prefix dict has been built successfully.
56 Building prefix dict from the default dictionary ...
57 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
58 Loading model cost 0.672 seconds.
59 Prefix dict has been built successfully.
60 Building prefix dict from the default dictionary ...
61 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
62 Loading model cost 0.619 seconds.
63 Prefix dict has been built successfully.
64 共用时: 22.58281373977661s
65 6进程处理完成
```



```
66 Building prefix dict from the default dictionary ...
67 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
68 Loading model cost 0.572 seconds.
69 Prefix dict has been built successfully.
70 Building prefix dict from the default dictionary ...
71 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
72 Loading model cost 0.580 seconds.
73 Prefix dict has been built successfully.
74 Building prefix dict from the default dictionary ...
75 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
76 Loading model cost 0.636 seconds.
77 Prefix dict has been built successfully.
78 Building prefix dict from the default dictionary ...
79 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
80 Loading model cost 0.630 seconds.
81 Prefix dict has been built successfully.
82 Building prefix dict from the default dictionary ...
83 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
84 Loading model cost 0.644 seconds.
85 Prefix dict has been built successfully.
86 Building prefix dict from the default dictionary ...
87 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
88 Loading model cost 0.648 seconds.
89 Prefix dict has been built successfully.
90 Building prefix dict from the default dictionary ...
91 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
92 Loading model cost 0.632 seconds.
93 Prefix dict has been built successfully.
94 Building prefix dict from the default dictionary ...
95 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
96 Loading model cost 0.653 seconds.
97 Prefix dict has been built successfully.
98 共用时: 25.392186641693115s
99 8进程处理完成
100 Building prefix dict from the default dictionary ...
101 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
102 Loading model cost 0.574 seconds.
103 Prefix dict has been built successfully.
104 Building prefix dict from the default dictionary ...
105 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
106 Loading model cost 0.578 seconds.
107 Prefix dict has been built successfully.
108 Building prefix dict from the default dictionary ...
109 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
110 Loading model cost 0.644 seconds.
111 Prefix dict has been built successfully.
112 Building prefix dict from the default dictionary ...
113 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
114 Loading model cost 0.661 seconds.
115 Prefix dict has been built successfully.
116 Building prefix dict from the default dictionary ...
```

```
117 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
118 Loading model cost 0.650 seconds.
119 Prefix dict has been built successfully.
120 Building prefix dict from the default dictionary ...
121 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
122 Loading model cost 0.621 seconds.
123 Prefix dict has been built successfully.
124 Building prefix dict from the default dictionary ...
125 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
126 Loading model cost 0.613 seconds.
127 Prefix dict has been built successfully.
128 Building prefix dict from the default dictionary ...
129 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
130 Loading model cost 0.643 seconds.
131 Prefix dict has been built successfully.
132 Building prefix dict from the default dictionary ...
133 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
134 Loading model cost 0.612 seconds.
135 Prefix dict has been built successfully.
136 Building prefix dict from the default dictionary ...
137 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
138 Loading model cost 0.600 seconds.
139 Prefix dict has been built successfully.
140 共用时: 28.55099058151245s
141 10进程处理完成
142 Building prefix dict from the default dictionary ...
143 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
144 Loading model cost 0.595 seconds.
145 Prefix dict has been built successfully.
146 Building prefix dict from the default dictionary ...
147 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
148 Loading model cost 0.586 seconds.
149 Prefix dict has been built successfully.
150 Building prefix dict from the default dictionary ...
151 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
152 Loading model cost 0.601 seconds.
153 Prefix dict has been built successfully.
154 Building prefix dict from the default dictionary ...
155 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
156 Loading model cost 0.617 seconds.
157 Prefix dict has been built successfully.
158 Building prefix dict from the default dictionary ...
159 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
160 Loading model cost 0.609 seconds.
161 Prefix dict has been built successfully.
162 Building prefix dict from the default dictionary ...
163 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
164 Loading model cost 0.603 seconds.
165 Prefix dict has been built successfully.
166 Building prefix dict from the default dictionary ...
167 Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
```



```

23     '''
24     jieba.load_userdict(dict_path)
25     f=open(dict_path,'r',encoding='utf8')
26     stop_words={line.strip() for line in f.readlines()}
27     f.close()
28     return stop_words
29 def Map(data,stop_words,result):
30     '''
31     Map进程读取文档并进行词频统计，返回该文本的词频统计结果
32     '''
33     count_dict={}
34     for line in data:
35         words=jieba.lcut(line)
36         for word in words:
37             if word in stop_words:
38                 continue
39             elif word not in count_dict:
40                 count_dict.update({word:0})
41             count_dict[word]+=1
42     result.append(count_dict)
43 def Reduce(result_lis,save_path):
44     '''
45     整合所有结果
46     '''
47     result_all={}
48     for result in tqdm(result_lis,desc='reducing...'):
49         for key,value in result.items():
50             if key not in result_all:
51                 result_all.update({key:0})
52             result_all[key]+=value
53     with open('test.csv','w',newline='',encoding='utf8') as f:
54         writer=csv.writer(f)
55         for row in result_all.items():
56             writer.writerow(row)
57 if __name__=='__main__':
58     file_path='D:/Project/Python/week11MapReduce/sohu_data.json'
59     dict_path='D:/Project/Python/week11MapReduce/stopwords_list.txt'
60     save_path='D:/Project/Python/week11MapReduce/result.csv'
61     #N=[psutil.cpu_count(False)]
62     N=[1,2,3,4,5,6,7,8,9,10,11,12]
63     data=ReadData(file_path)
64     stop_words=LoadStopwords(dict_path)
65     N_time=[]
66     data=data[:30000]#减小数据量
67     size=len(data)
68     for n in N:
69         p_list=[]
70         m=Manager()
71         result=m.list([])
72         for i in range(n):#创建CPU内核数个进程
73             p=Process(target=Map,args=(data[int(size/n*i):int(size/n*(i+1))])

```

```
74         p_list.append(p)
75     start_time=time.time()
76     for p in p_list:
77         p.start()#启动进程
78     for p in p_list:
79         p.join()#阻滞主进程
80     t=time.time()-start_time
81     print('共用时: {}s'.format(t))#测试总用时
82     N_time.append(t)
83     print('{}进程处理完成'.format(n))
84 Reduce(result,save_path)
85 plt.plot(N,N_time)
86 plt.show()
```