现代程序设计第11周作业

谢奕飞 20377077

ReadData函数

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

LoadStopwords函数

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

Map函数

```
def Map(data,stop_words,result):
    '''
    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
    '''
    count_dict={}
    for line in data:
        words=jieba.lcut(line)
        for word in words:
            if word in stop_words:
                continue
            elif word not in count_dict:
                count_dict.update({word:0})
            count_dict[word]+=1
```

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

Reduce函数

```
def Reduce(result_lis,save_path):
    '''
    整合所有结果
    '''
    result_all={}
    for result in tqdm(result_lis,desc='reducing...'):
        for key,value in result.items():
            if key not in result_all:
                result_all.update({key:0})
            result_all[key]+=value
    with open('test.csv','w',newline='',encoding='utf8') as f:
        writer=csv.writer(f)
        for row in result_all.items():
            writer.writerow(row)
```

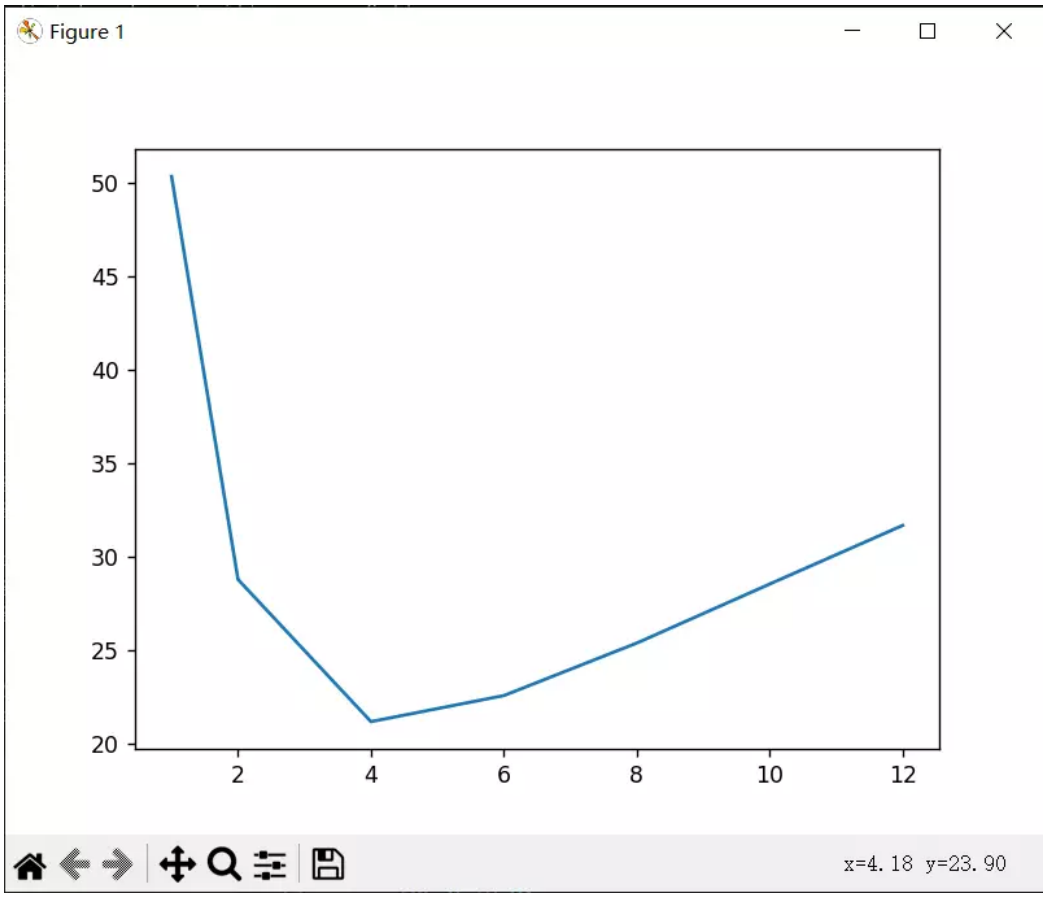
main函数

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

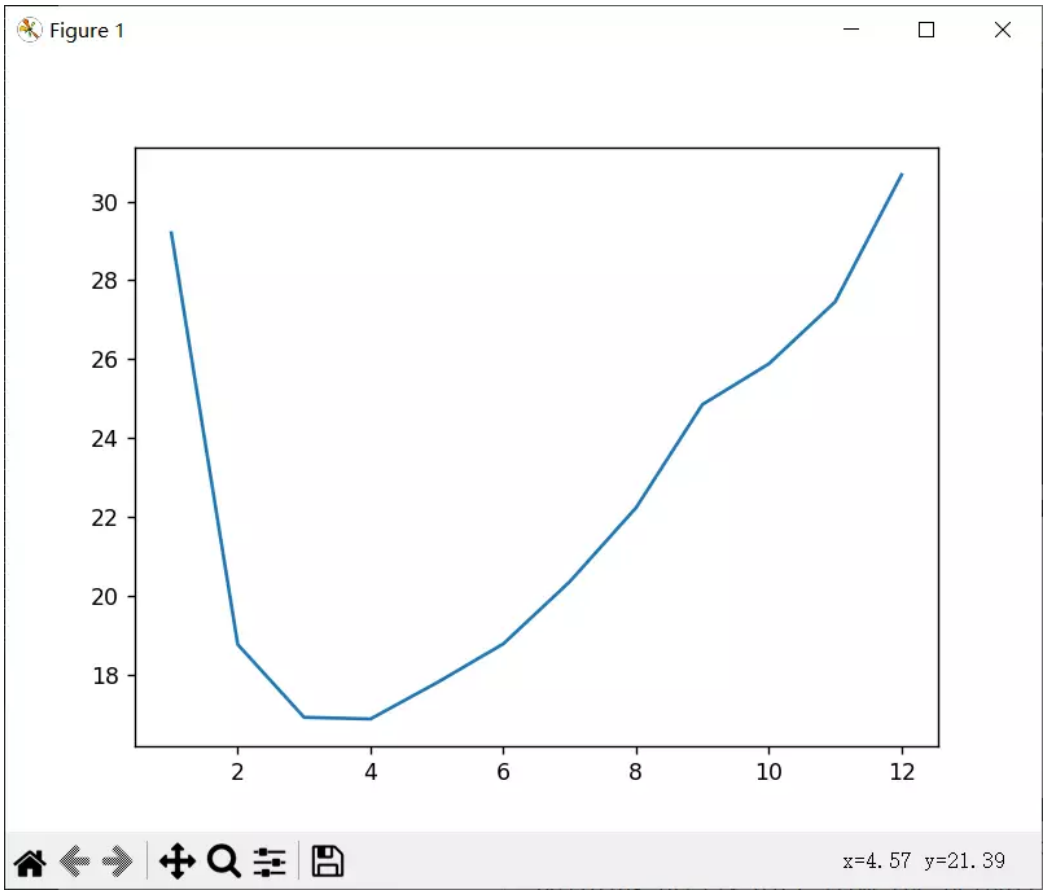
```
    plt.plot(N,N_time)
    plt.show()
```

运行结果

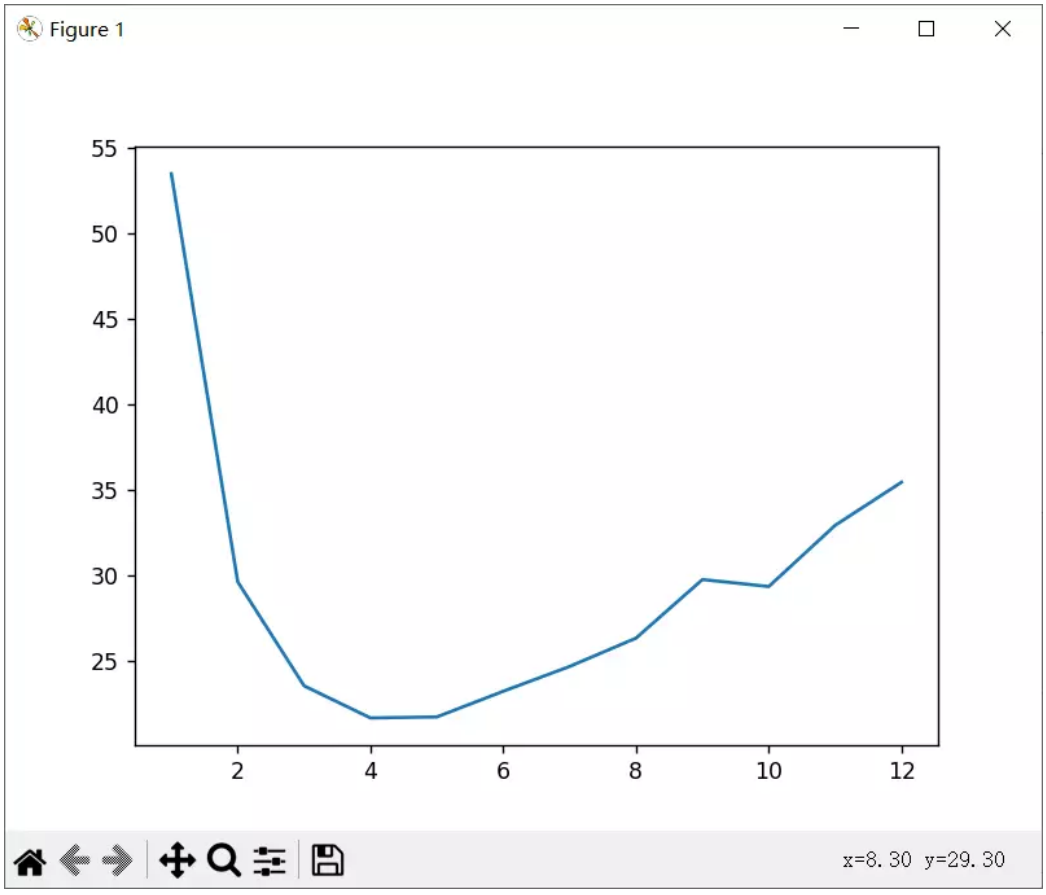
```
N=[1,2,4,6,8,10,12]
data=data[:10000]#减小数据量
```



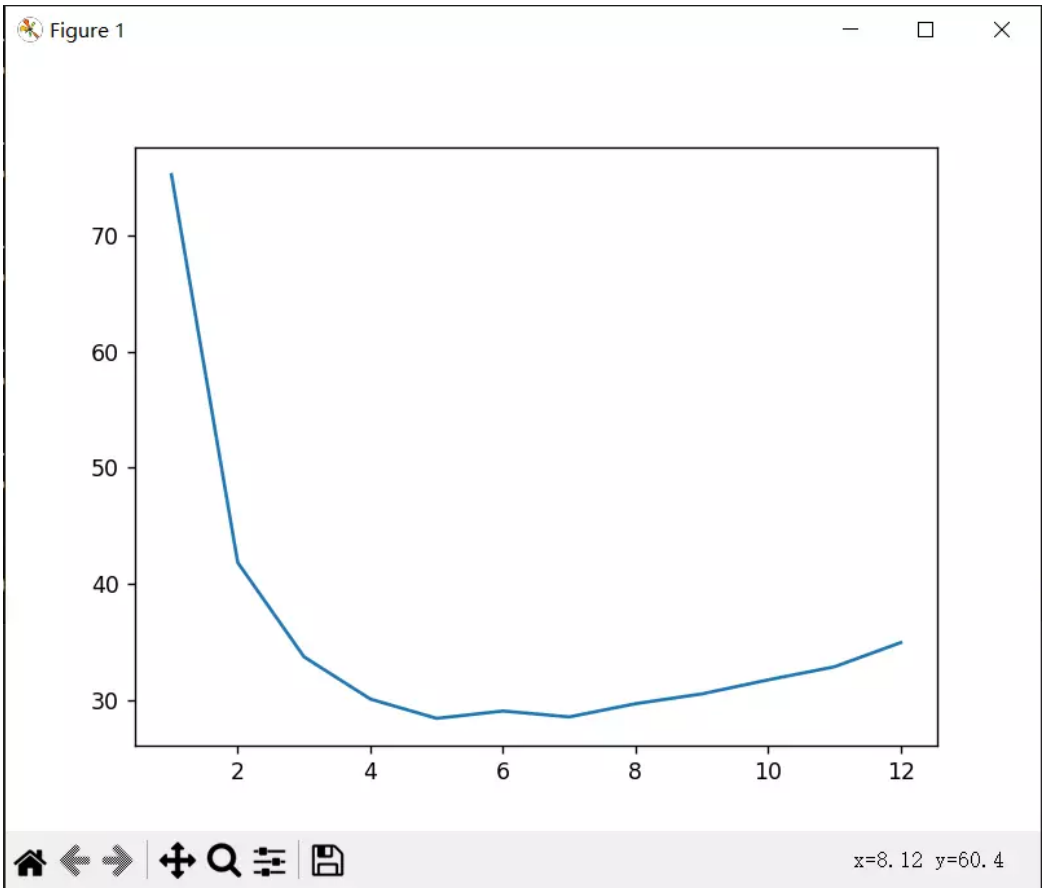

```
N=[1,2,3,4,5,6,7,8,9,10,11,12]
data=data[:5000]#减小数据量
```



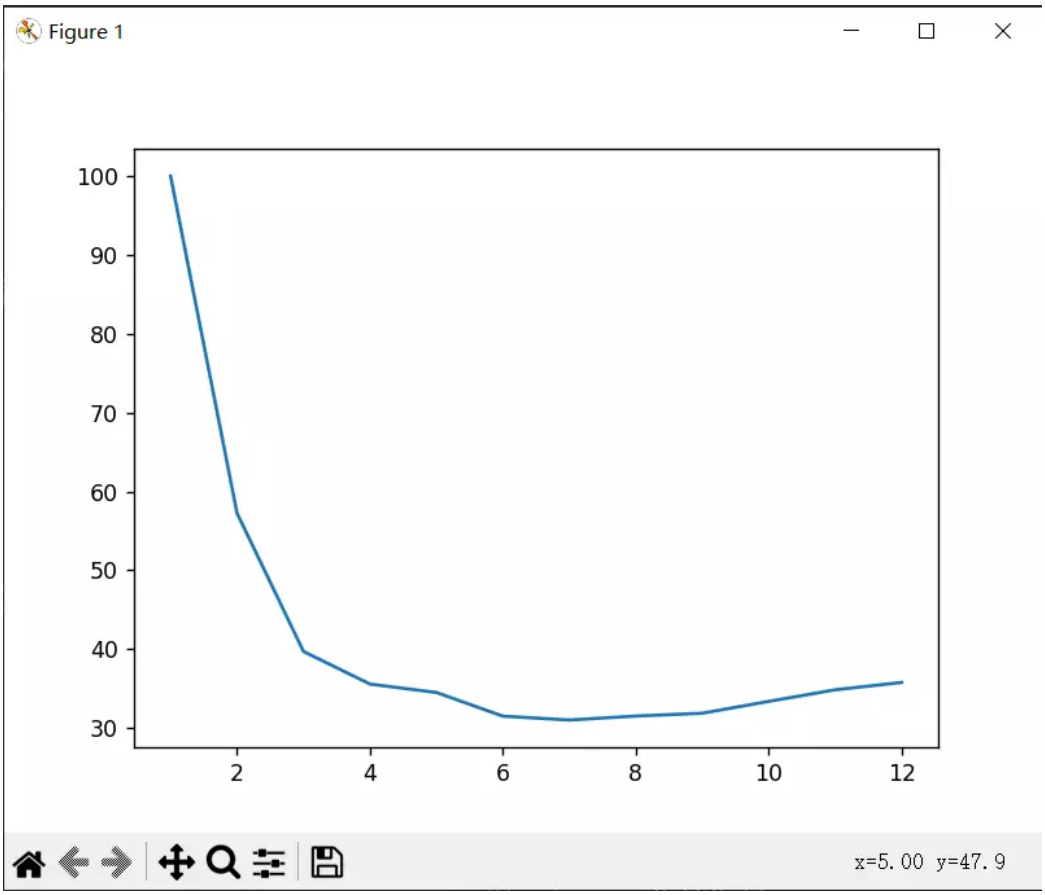

```
N=[1,2,3,4,5,6,7,8,9,10,11,12]
data=data[:10000]#减小数据量
```



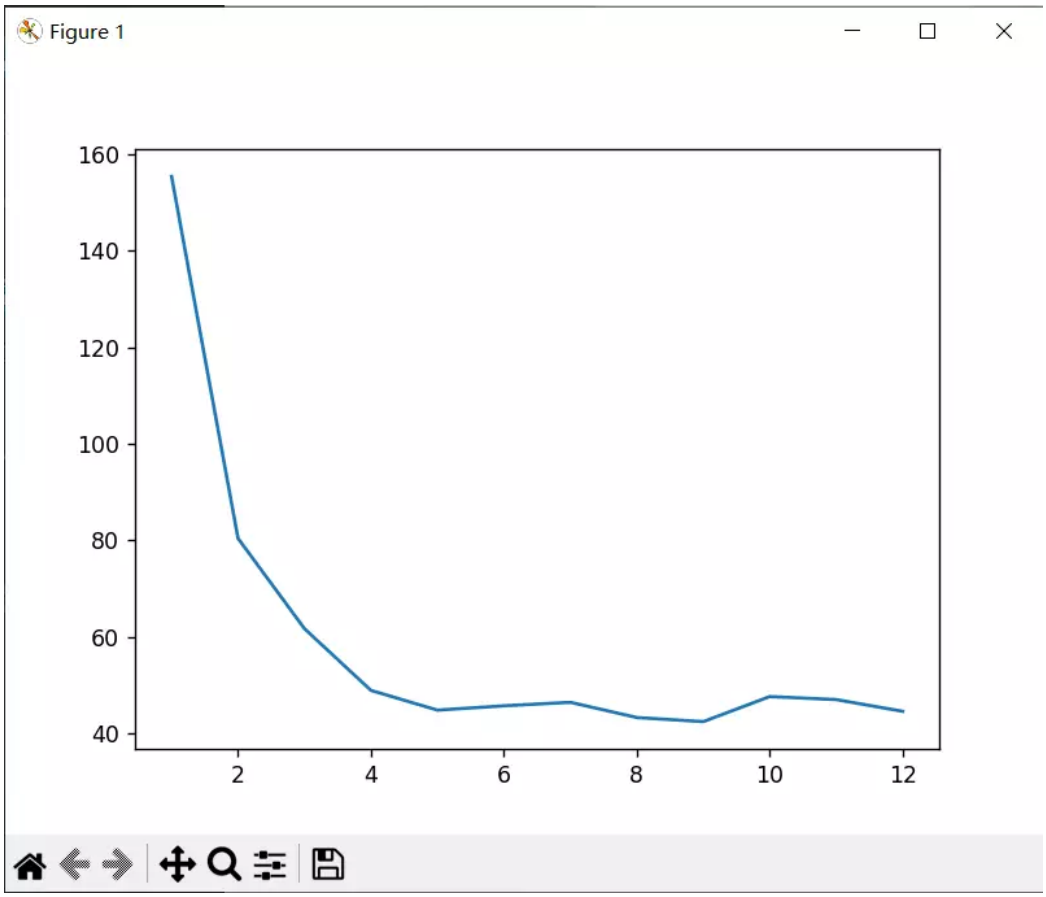

```
N=[1,2,3,4,5,6,7,8,9,10,11,12]
data=data[:15000]#减小数据量
```




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

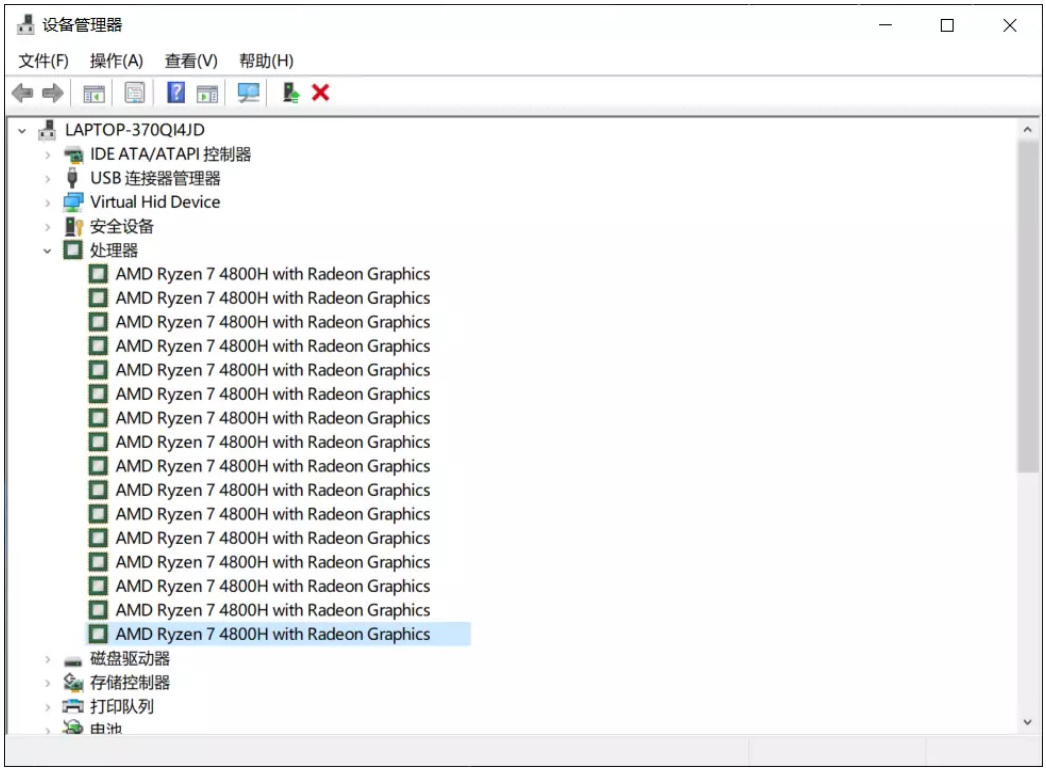



```
N=[1,2,3,4,5,6,7,8,9,10,11,12]
data=data[:30000]#减小数据量
```



分析

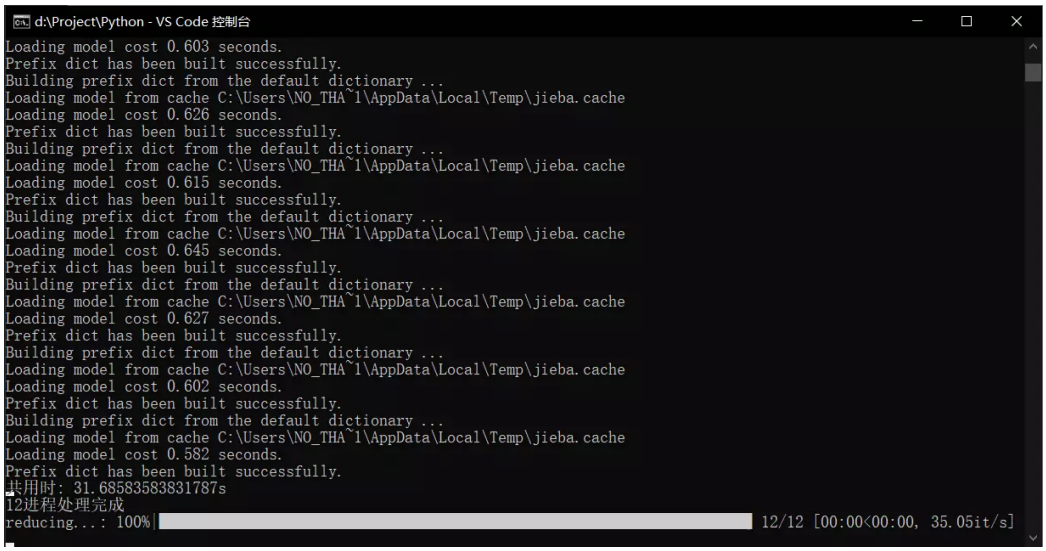
我的CPU是AMD R7-4800H，8核16线程。



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



详细输出

以

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

为例

```
reading...: 100%|████████████████████████████████████████████████████████████|
1245835/1245835 [00:01<00:00, 693798.41it/s]
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.553 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 0.564 seconds.
Prefix dict has been built successfully.
共用时: 50.357043504714966s
1进程处理完成
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.572 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 0.567 seconds.
Prefix dict has been built successfully.
共用时: 28.794530391693115s
2进程处理完成
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.586 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 0.585 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 0.621 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 0.620 seconds.
Prefix dict has been built successfully.
共用时: 21.189520120620728s
4进程处理完成
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.593 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 0.590 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 0.604 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 0.611 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 0.672 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 0.619 seconds.
Prefix dict has been built successfully.
共用时: 22.58281373977661s
6进程处理完成
```

```
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.572 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 0.580 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 0.636 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 0.630 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 0.644 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 0.648 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 0.632 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 0.653 seconds.
Prefix dict has been built successfully.
共用时: 25.392186641693115s
8进程处理完成
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.574 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 0.578 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 0.644 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 0.661 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 0.650 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 0.621 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 0.613 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 0.643 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 0.612 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 0.600 seconds.
Prefix dict has been built successfully.
共用时: 28.55099058151245s
10进程处理完成
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.595 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 0.586 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 0.601 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 0.617 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 0.609 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 0.603 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]
```

附录–完整代码

```python
import json
from tqdm import tqdm
import jieba
import time
import csv
from multiprocessing import Process,Manager
import psutil
from matplotlib import pyplot as plt

def ReadData(file_path):
    '''
    读取json文件,返回行的列表
    '''
    result=[]
    with open(file_path,encoding='utf8') as f:
        data=json.load(f)
        for line in tqdm(data,desc='reading...'):
            result.append(line.get('content'))
    return result
def LoadStopwords(dict_path):
    '''
    jieba载入分词库,返回分词集合
```

```
    '''
    jieba.load_userdict(dict_path)
    f=open(dict_path,'r',encoding='utf8')
    stop_words={line.strip() for line in f.readlines()}
    f.close()
    return stop_words
def Map(data,stop_words,result):
    '''
    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
    '''
    count_dict={}
    for line in data:
        words=jieba.lcut(line)
        for word in words:
            if word in stop_words:
                continue
            elif word not in count_dict:
                count_dict.update({word:0})
            count_dict[word]+=1
    result.append(count_dict)
def Reduce(result_lis,save_path):
    '''
    整合所有结果
    '''
    result_all={}
    for result in tqdm(result_lis,desc='reducing...'):
        for key,value in result.items():
            if key not in result_all:
                result_all.update({key:0})
            result_all[key]+=value
    with open('test.csv','w',newline='',encoding='utf8') as f:
        writer=csv.writer(f)
        for row in result_all.items():
            writer.writerow(row)
if __name__=='__main__':
    file_path='D:/Project/Python/week11MapReduce/sohu_data.json'
    dict_path='D:/Project/Python/week11MapReduce/stopwords_list.txt'
    save_path='D:/Project/Python/week11MapReduce/result.csv'
    #N=[psutil.cpu_count(False)]
    N=[1,2,3,4,5,6,7,8,9,10,11,12]
    data=ReadData(file_path)
    stop_words=LoadStopwords(dict_path)
    N_time=[]
    data=data[:30000]#减小数据量
    size=len(data)
    for n in N:
        p_list=[]
        m=Manager()
        result=m.list([])
        for i in range(n):#创建CPU内核数个进程
            p=Process(target=Map,args=(data[int(size/n*i):int(size/n*(i+1))]
```

```
            p_list.append(p)
        start_time=time.time()
        for p in p_list:
            p.start()#启动进程
        for p in p_list:
            p.join()#阻滞主进程
        t=time.time()-start_time
        print('共用时: {}s'.format(t))#测试总用时
        N_time.append(t)
        print('{}进程处理完成'.format(n))
    Reduce(result,save_path)
    plt.plot(N,N_time)
    plt.show()
```