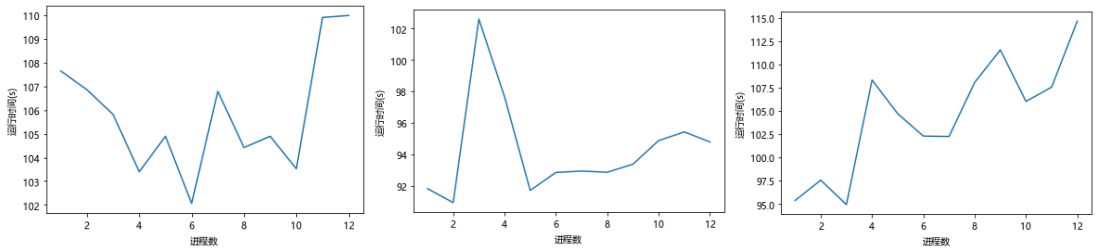
注：因 sogouCS 数据集过大，所以用某旅游评论数据集代替。

```python
import time
import jieba
import pandas as pd
import matplotlib.pyplot as plt
from pathlib import Path
from multiprocessing import Process, Pool, Manager

plt.rcParams['font.sans-serif']=['Microsoft YaHei']  # 显示中文



def map_process(q, path):
    # 读取文本数据
    data = pd.read_csv(path).iloc[:, 0]
    tokens = data.apply(lambda x: [i for i in jieba.lcut(x)], 1)
    # 词频统计
    all_words =  (x for l in tokens for x in l)
    count = pd.Series(all_words).value_counts()
    q.put(count)

def reduce_process(q):
    flag = 1
    while True:
        count = q.get()
        if type(count).__name__ == 'NoneType':
            break
        else:
            if flag:
                flag = 0
                C = count
            else:
                for i in count.index:
                    if i in C.index:
                        C[i] += count[i]
                    else:
                        C[i] = count[i]
    C.to_csv('word_count.csv')
```

```python
if __name__ == '__main__':
    run_time = []
    for n in range(1, 13):
        start = time.time()  # 程序开始时间
        q = Manager().Queue()
        reduce = Process(target = reduce_process, args = (q,))
        reduce.daemon = True
        reduce.start()
        maps = Pool(n)
        for path in Path('data/').glob('*.*'):
            maps.apply_async(map_process, args=(q, path))
        maps.close()
        maps.join()
        q.put(None)  # 发出结束信号
        reduce.join()  # 等map结束后再结束reduce
        end = time.time()  # 程序结束时间
        run_time.append(end - start)
    plt.plot(list(range(1, 13)), run_time)
    plt.xlabel('进程数')
    plt.ylabel('运行时间(s)')
```

为消除运行结果的随机性，多次运行结果如下：



首先，可以看到每次运行的结果差异较大。设备 CPU 核数为 12，可能是电脑后台运行程序占用内存空间导致每次运行结果不一致，尤其是当进程数上升时运行时间反而上升。

总体上，当进程数为 1，4 或 6 时运行时间较短，效率较高，说明多进程的运行效率并不是进程数越多越好，受进程间通信时间的影响，运行效率有一个局部最优点。