# 多进程，多线程的数据融合策略

**技术背景：**由于数据数据集的数据量大，且前期数据融合过程中，对于原始数据只有读情况，即避免了死锁情况，适合多进程，多线程加速，写过程存在冲突需要解决。多进程：multiprocessing、多线程：threading

**方案：根据aq\_df中station\_id建立任务池，对于每个id调用多进程**

# print("主进程的PID为:{0}".format(os.getpid())) # 输出当前进程的PID。

processPool = Pool(50) # 使用Pool创建进程池，并定义最大可同时执行多少的进程。

for x in range(35): # 程序一共会生成35个进程，如果进程池数小于所有执行进程数，当进程结束后，再开始执行新进程。

processPool.apply\_async(testMuti, args=(x,)) # 申请异步进程。

processPool.close() # 关闭进程池，使之不会再有新进程开启。

processPool.join() # 使当前主进程等待所有子进程，在执行完毕后再结束主进程。

print("主进程结束。")

