

【Python】平行運算範例

大家好~

最近稍微研究了一下python的平行運算，

我們在分析時，常會透過loop計算大量的資料，

例如蒐集每個月顧客的資料，再做後續分析，

然而若是一個月一個月去跑，

可能會耗費大量的時間，

此時我們可以將不同的月份資料，

分散到不同的CPU上面，

平行運算後，

再結合在一起，

提升運算效率。

如下程式碼範例，

同樣是計算三個月的資料，

若以一般的loop去跑，需要耗費1分32秒，而採用平行運算，僅需28秒，

執行時間： 0:01:32.765701
資料筆數： 5979335

	date_ym	cust_no
0	202210	239453131302139853057514
1	202210	219533230311539643932023

(一般loop) →

執行時間： 0:00:28.774827
資料筆數： 5979335


	date_ym	cust_no
0	202210	2384E3231321539823251498
1	202210	206413231301139573832471

(平行運算)

推薦大家有空可以玩玩看，

速度UPUP(但相對的，其實就是在燒CPU，建議別開太大)

【參考程式碼路徑】

網路 > > DataMart > 智能數據應用小組 > 03_CODE參考 > python			
名稱	修改日期	類型	大小
 平行運算範例.ipynb	2023/8/22 下午 04:51	IPYNB 檔案	9 KB

【程式碼解釋】

引入套件

```
from joblib import Parallel, delayed
import joblib
```

範例說明: 透過get_pure_cc，可以產製一個月的純卡友資料，預計產出三個月的資料再合併

```
month_list = c81146.get_month_list(start_m, end_m)
Parallel_results = Parallel(n_jobs = 4)(delayed(func)(i) for i in month_list)
result = pd.concat(Parallel_results, ignore_index=True)
```

如上圖所示，先透過C81146.get_month_list(start_m, end_m) 產出月份的list (結果為: ['202210', '202211', '202212'])

第二行code 即為平行運算，可以透過n_jobs = 4 設定採用4個CPU (若寫為-1則代表CPU全開，建議別開太滿)

func即為get_pure_cc (因寫在裝飾器裡面，所以才用func代替)，i則是透過for迴圈丟入不同月份

第三行code將平行運算後的結果轉換為dataframe，以利未來作業。

PS: 範例程式碼有採用裝飾器，此裝飾器(combine_all_month_data)的作用即為以平行運算計算不用月份的資料並合併，可以先稍微了解一下~

PS2: 可以透過 joblib.cpu_count() 確認有多少CPU可以使用

```
joblib.cpu_count()
```