

## Tiempos de ejecucion en secuencial

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from seaborn import load_dataset
```

```
In [3]: datos = pd.read_csv("seq.txt")
datos.head()
```

Out[3]:

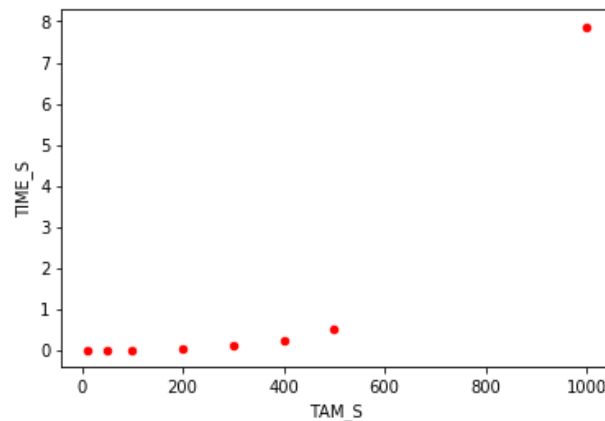
	TAM_S	TIME_S
0	10	0.000007
1	50	0.000702
2	100	0.003615
3	200	0.028920
4	300	0.103045

```
In [4]: datSeq = datos.groupby(["TAM_S"]).mean()
datSeq.reset_index(level=0,inplace=True)
datSeq
```

Out[4]:

	TAM_S	TIME_S
0	10	0.000005
1	50	0.000675
2	100	0.004015
3	200	0.028736
4	300	0.098529
5	400	0.241374
6	500	0.500766
7	1000	7.876333

```
In [5]: datSeq.plot(kind='scatter',x='TAM_S',y='TIME_S',color='red')
plt.show()
```



## Tiempos de ejecucion paralelo

```
In [6]: datos_p = pd.read_csv("pthread.txt")
datos_p.head()
```

Out[6]:

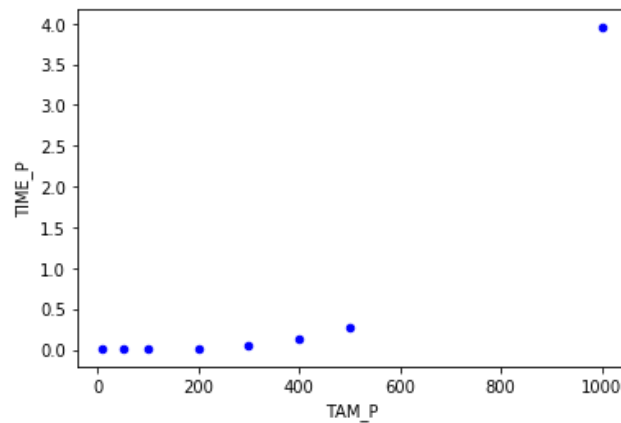
	TAM_P	TIME_P
0	10	0.000371
1	50	0.001296
2	100	0.003944
3	200	0.018586
4	300	0.054909

```
In [7]: datParl = datos_p.groupby(["TAM_P"]).mean()
datParl.reset_index(level=0,inplace=True)
datParl
```

Out[7]:

	TAM_P	TIME_P
0	10	0.000630
1	50	0.001879
2	100	0.003751
3	200	0.018815
4	300	0.057465
5	400	0.138173
6	500	0.281198
7	1000	3.958512

```
In [8]: datParl.plot(kind='scatter',x='TAM_P',y='TIME_P',color='blue')
plt.show()
```



## Speedup

```
In [9]: datosMerge = pd.merge(left = datParl, left_index = False, left_on='TAM_P',
                               right=datSeq, right_index=False, right_on = 'TAM_S')
datosMerge
```

Out[9]:

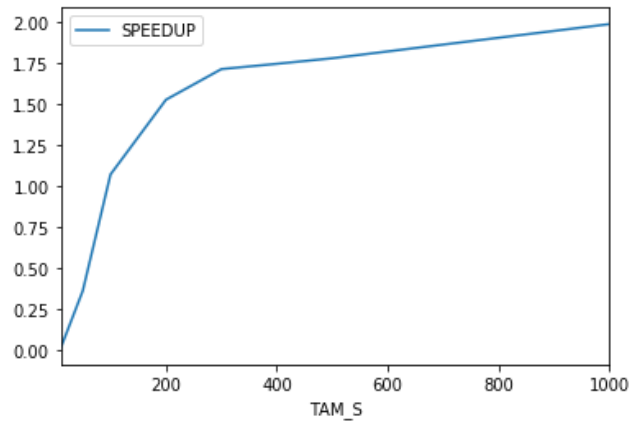
	TAM_P	TIME_P	TAM_S	TIME_S
0	10	0.000630	10	0.000005
1	50	0.001879	50	0.000675
2	100	0.003751	100	0.004015
3	200	0.018815	200	0.028736
4	300	0.057465	300	0.098529
5	400	0.138173	400	0.241374
6	500	0.281198	500	0.500766
7	1000	3.958512	1000	7.876333

```
In [10]: datosMerge['SPEEDUP'] = datosMerge["TIME_S"] / datosMerge["TIME_P"]
datosMerge
```

```
Out[10]:
```

	TAM_P	TIME_P	TAM_S	TIME_S	SPEEDUP
0	10	0.000630	10	0.000005	0.007621
1	50	0.001879	50	0.000675	0.359370
2	100	0.003751	100	0.004015	1.070515
3	200	0.018815	200	0.028736	1.527332
4	300	0.057465	300	0.098529	1.714598
5	400	0.138173	400	0.241374	1.746895
6	500	0.281198	500	0.500766	1.780828
7	1000	3.958512	1000	7.876333	1.989721

```
In [11]: ax = plt.gca()
datosMerge.plot(kind='line', x='TAM_S', y='SPEEDUP', ax=ax)
plt.show()
```



## Conclusiones

- Es notorio que los tiempos de ejecución de los algoritmos en paralelo rebozan por mucho la ejecución secuencial. Para todos los datos en los documentos txt se obtiene mejores tiempos con los algoritmos paralelos desde el inicio de la toma de datos.
- Se evidencia que una maquina con mejor nivel de procesamiento refleja una resolución a la cantidad de tareas mucho mas rapida que en una maquina con menos nivel de procesamiento.
- La diferencia en la aceleración entre estos algoritmos de paralelización parece que en algún momento tiende a un valor fijo. Se debe recordar que la maquina usada para la toma de los datos no es de uso específico y esto puede introducir ruido en los tiempos de ejecución.