

# Serial vs parallel

python

Meelis Utt

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import time

def f(x):
    return(x**2 + x**4 + np.sin(x) + np.cos(x) + x**25)

analytical = 613/390 + np.sin(1) - np.cos(1)

def MC(n):
    pyunif = np.random.uniform(0,1,n)
    EX = np.mean(f(pyunif))
    error = np.abs(EX-analytical)
    return(EX,error)

N = 1000000 * np.array([1,2.5,5,7.5,10], dtype=int)
EX = []
error = []
type = []
walltime = []
for n in N:
    start = time.time()
    ex,err = MC(int(n))
    end = time.time()
    EX.append(ex);error.append(err);type.append("MCser");walltime.append(end-start)

# MCpar results
rm -f resultsPar.csv
for p in 1 2 4 5 10 25 50 100
do
    for n in 1000000 2500000 5000000 7500000 10000000
    do
        mpirun -n $p --hostfile hostfile python MCpar.py $n >> resultsPar.csv
    done
done

# dict = {"n": N, "EX": EX, "error": error, "type": type, "walltime": walltime}
dict = {"n": N, "EX": EX, "error": error, "type": type, "walltime": walltime}

dfSer = pd.DataFrame(dict)
print(dfSer)
```

```
##          n          EX          error          type          walltime
```

```
## 0 1000000 1.873666 0.000702 MCser 0.108748
## 1 2000000 1.872993 0.000030 MCser 0.227737
## 2 5000000 1.873553 0.000589 MCser 0.546811
## 3 7000000 1.872971 0.000007 MCser 0.749277
## 4 10000000 1.873204 0.000240 MCser 1.063224
```

```
dfPar = pd.read_csv("resultsPar.csv",header=0,names=["n","EX","error","type","walltime","processes"])
print(dfPar)
```

##	n	EX	error	type	walltime	processes
## 0	2500000	1.873661	0.000697	MCpar	0.269072	1
## 1	5000000	1.872906	0.000057	MCpar	0.538868	1
## 2	7500000	1.873263	0.000299	MCpar	0.801892	1
## 3	10000000	1.873060	0.000097	MCpar	1.091981	1
## 4	1000000	1.872625	0.000339	MCpar	0.067745	2
## 5	2500000	1.872784	0.000179	MCpar	0.155509	2
## 6	5000000	1.873301	0.000337	MCpar	0.311301	2
## 7	7500000	1.872664	0.000299	MCpar	0.466155	2
## 8	10000000	1.873007	0.000043	MCpar	0.702686	2
## 9	1000000	1.874304	0.001341	MCpar	0.062437	4
## 10	2500000	1.873101	0.000138	MCpar	0.144517	4
## 11	5000000	1.872350	0.000613	MCpar	0.254579	4
## 12	7500000	1.872904	0.000059	MCpar	0.393794	4
## 13	10000000	1.873565	0.000602	MCpar	0.497969	4
## 14	1000000	1.873524	0.000560	MCpar	0.184178	5
## 15	2500000	1.872767	0.000197	MCpar	0.260226	5
## 16	5000000	1.872800	0.000163	MCpar	0.357716	5
## 17	7500000	1.873081	0.000117	MCpar	0.490899	5
## 18	10000000	1.873209	0.000245	MCpar	0.460575	5
## 19	1000000	1.873321	0.000358	MCpar	0.086715	10
## 20	2500000	1.872030	0.000933	MCpar	0.232074	10
## 21	5000000	1.873349	0.000386	MCpar	0.327013	10
## 22	7500000	1.872448	0.000516	MCpar	0.470002	10
## 23	10000000	1.872965	0.000002	MCpar	0.551064	10
## 24	1000000	1.874251	0.001288	MCpar	0.261662	25
## 25	2500000	1.873399	0.000435	MCpar	0.258084	25
## 26	5000000	1.873169	0.000205	MCpar	0.485729	25
## 27	7500000	1.872729	0.000234	MCpar	0.589378	25
## 28	10000000	1.873255	0.000291	MCpar	0.871043	25
## 29	1000000	1.873218	0.000254	MCpar	0.406218	50
## 30	2500000	1.873259	0.000295	MCpar	0.694380	50
## 31	5000000	1.873580	0.000616	MCpar	0.549333	50
## 32	7500000	1.873074	0.000110	MCpar	0.810069	50
## 33	10000000	1.873478	0.000514	MCpar	1.547999	50
## 34	1000000	1.873019	0.000055	MCpar	0.800761	100
## 35	2500000	1.872278	0.000685	MCpar	0.338554	100
## 36	5000000	1.873280	0.000317	MCpar	1.313040	100
## 37	7500000	1.873019	0.000056	MCpar	0.871543	100
## 38	10000000	1.872892	0.000072	MCpar	1.044397	100

```
ax = plt.gca()
dfSer.plot(kind='line',x='n',y='walltime',style='.-',rot=0,label='Serial')

dfPar.groupby('processes').plot(kind='line',x='n',y='walltime',
style='.-',rot=0,ax=plt.gca(),label="Parallel "+str('processes'))
```

```
# plt.legend(bbox_to_anchor=(1.25,0.75), bbox_transform=ax.transData)
```

```
## processes
```

```
## 1 AxesSubplot(0.125,0.11;0.775x0.77)
```

```
## 2 AxesSubplot(0.125,0.11;0.775x0.77)
```

```
## 4 AxesSubplot(0.125,0.11;0.775x0.77)
```

```
## 5 AxesSubplot(0.125,0.11;0.775x0.77)
```

```
## 10 AxesSubplot(0.125,0.11;0.775x0.77)
```

```
## 25 AxesSubplot(0.125,0.11;0.775x0.77)
```

```
## 50 AxesSubplot(0.125,0.11;0.775x0.77)
```

```
## 100 AxesSubplot(0.125,0.11;0.775x0.77)
```

```
## dtype: object
```

```
plt.legend(['Serial','Parallel P=1','Parallel P=2','Parallel P=4',  
           'Parallel P=5','Parallel P=10','Parallel P=25',  
           'Parallel P=50','Parallel P=100'],loc='upper left')
```

```
plt.show()
```

```
# dfPar.join(dfSer,on="n",lsuffix=".par",rsuffix=".ser")
```

```
# TODO: plot relative speedups
```

