Analysis_Q1

September 16, 2020

1 Analysis for question 1

Raj Patil
CS18BTECH11039

[14]:
```
from time import perf_counter as pc
from dis import dis as da #dis-assembler
from matplotlib import pyplot as plt
import numpy as np
```

2 Performance analysis

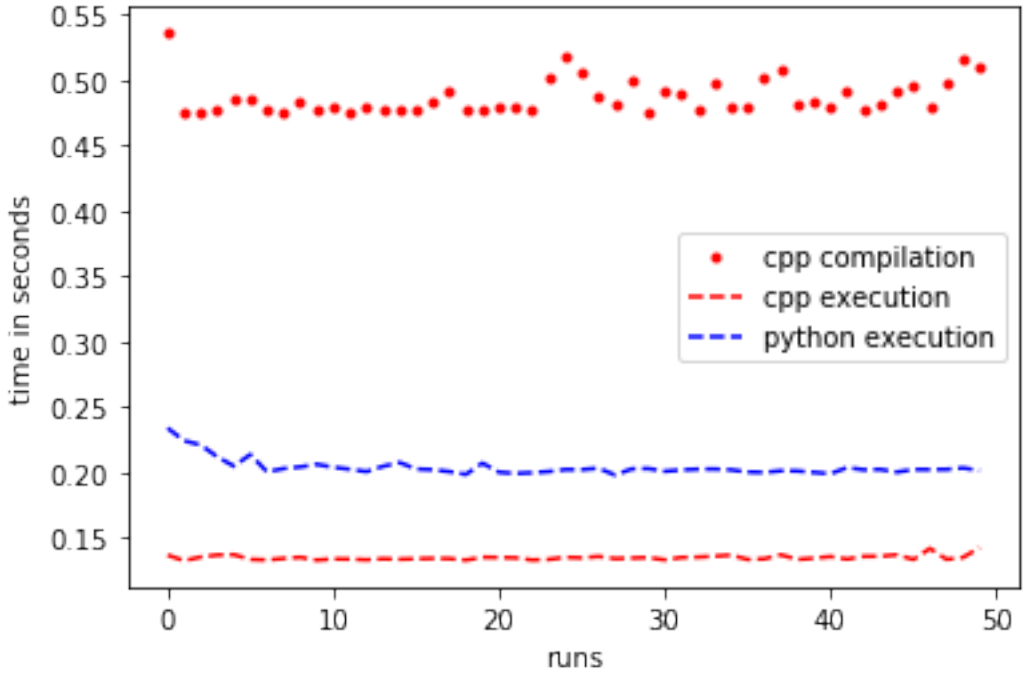
[68]:
```
compile_times_cpp=[]
execution_times_cpp=[]

for i in range(50):
    t0_cpp = pc()
    !clang++ test1.cpp
    t_cpp_compile=pc()-t0_cpp
    !./a.out > /dev/null
    t_cpp_all = pc()-t0_cpp
    compile_times_cpp.append(t_cpp_compile)
    execution_times_cpp.append(t_cpp_all-t_cpp_compile)
```

[47]:
```
execution_times_py=[]
for i in range(50):
    t0_py = pc()
    !python test1.py > /dev/null
    t_py = pc() - t0_py
    execution_times_py.append(t_py)
```

[69]:
```
plt.plot(range(50),compile_times_cpp,"r.",label="cpp compilation")
plt.plot(range(50),execution_times_cpp,"r--",label="cpp execution")
plt.plot(range(50),execution_times_py,"b--",label="python execution")
plt.xlabel("runs")
```

```
plt.ylabel("time in seconds")
plt.legend()
plt.show()
```



2.1 mean speedup:

cpp execution over python execution

[78]:
```
(np.asarray(execution_times_py)/np.asarray(execution_times_cpp)).mean()
```

[78]:
```
1.5119598619465746
```