Seung Ki Lee

CSE3342 - HW9

Code for MultiThread

```
71 def partitionWorkMultiThread(nComputation, nWorkers):
72     #Set up a list
73     threadList = []
74     computationsPerWorker = nComputations / nWorkers
75
76
77     #create and add threads to threadllist
78     for x in range (0, nWorkers):
79         start = computationsPerWorker * x
80         stop = computationsPerWorker * (x + 1)
81         t = threading.Thread(target=dowork, args=(start,stop))
82         threadList.append(t)
83
84     #iterate over threadlist to launch
85     for y in threadList:
86         y.start()
87
88     #iterate over the same list to have them wait
89     for h in threadList:
90         h.join()
91 # DO timing for partitionWorkOneThread(nComputations, nWorkers)
92 # The final total should be the same if done correctly
93 bigTotal = 0
94 nWorkers = 100
95
96 startTime = time()
97 partitionWorkMultiThread(nComputations,nWorkers)
98 endTime = time()
99 timeMultiThreadWorkers = endTime - startTime
100 print "MultiThread Thread (Workers) :  SUM= %06f  Time=%07f " % (bigTotal, timeMultiThreadWorkers)
```

The results were weird. The Multithread with start and join in the separate loop took on average longer time

```
In [13]: runfile('C:/Users/lg/Downloads/hw8Timing.py', wdir='C:/Users/
lg/Downloads')
Single Thread           :  SUM= 4998925.462873  Time=6.509037
Single Thread (Workers) :  SUM= 4998925.462873  Time=6.612939
MultiThread Thread (Workers) :  SUM= 4998925.462873  Time=14.731164

In [14]: runfile('C:/Users/lg/Downloads/hw8Timing.py', wdir='C:/Users/
lg/Downloads')
Single Thread           :  SUM= 4998925.462873  Time=6.850322
Single Thread (Workers) :  SUM= 4998925.462873  Time=7.077531
MultiThread Thread (Workers) :  SUM= 4998925.462873  Time=15.791198

In [15]: runfile('C:/Users/lg/Downloads/hw8Timing.py', wdir='C:/Users/
lg/Downloads')
Single Thread           :  SUM= 4998925.462873  Time=6.928316
Single Thread (Workers) :  SUM= 4998925.462873  Time=7.655081
MultiThread Thread (Workers) :  SUM= 4998925.462873  Time=18.534336

In [16]: runfile('C:/Users/lg/Downloads/hw8Timing.py', wdir='C:/Users/
lg/Downloads')
Single Thread           :  SUM= 4998925.462873  Time=6.789522
Single Thread (Workers) :  SUM= 4998925.462873  Time=6.920669
MultiThread Thread (Workers) :  SUM= 4998925.462873  Time=15.468173
```

On the other hand, putting the start and join in the same loop presented result where MultiThread was at same speed as single Thread

```
In [17]: runfile('C:/Users/lg/Downloads/hw8Timing.py', wdir='C:/Users/
lg/Downloads')
Single Thread           :  SUM= 4998925.462873  Time=6.425803
Single Thread (Workers) :  SUM= 4998925.462873  Time=6.708687
MultiThread Thread (Workers) :  SUM= 4998925.462873  Time=6.729707
```

I presume that by putting start and stop in the same loop, it executes and finishes the thread before other threads are instantiated and essentially does not have the wait – effectively turning into a single thread work.