**Assignment A- Thinking in Parallel**

**By: Monica Dabas**

Q1. What native features of Python support parallel programming?

Concurrent program can be written using async and yield. Since these are concurrent programs, the tasks will run independently of each other but on a single thread.

PS: All Python implementations come prepackaged with multiprocessing module which provides parallel programming features.

Q2. What native features of R support parallel programming?

The lapply function in R supports parallel programming. For e.g. if we want to apply k-means clustering on 1 million data points on a machine with 5 cores, we can divide the data points into 4 subsets and lapply can use 4 cores to compute the means for each subset.

Q3. What Python libraries support parallel programming?

Python has a number of libraries to support parallel programming like PP (Parallel Python), multiprocessing, Mpi4Py, RealThreading: The multi-processing module for Python 3.5 and many others as mentioned on the below link. <https://wiki.python.org/moin/ParallelProcessing>

Q4. What R libraries (packages) support parallel programming?

R has a large number of libraries/ packages that support parallel programming. Some of those are: parallel, snowfall, doMC: Foreach Parallel Adaptor for 'parallel', foreach: Provides Foreach Looping Construct for R, multicore: R package for parallel processing of R code, Microsoft R Open (earlier called Revolution R Open (RRO)), BLAS/LAPACK, Rmpi, packages in CRAN Task View: High-Performance and Parallel Computing with R

(<https://cran.r-project.org/web/views/HighPerformanceComputing.html>),

Q5. Can GPUs be used with Python and R? If yes, then how?

GPU’s can be used in Python and R by using functionalities from following modules/libraries.

With R: [gputools](https://cran.r-project.org/web/packages/gputools/index.html) package (A Few GPU Enabled Functions) with Nvidia CUDA toolkit

[cudaBayesreg](https://cran.r-project.org/web/packages/cudaBayesreg/index.html), [gcbd](https://cran.r-project.org/web/packages/gcbd/index.html), [OpenCL](https://cran.r-project.org/web/packages/OpenCL/index.html), [HiPLARM](https://cran.r-project.org/web/packages/HiPLARM/index.html), [permGPU](https://cran.r-project.org/web/packages/permGPU/index.html), [gmatrix](https://cran.r-project.org/web/packages/gmatrix/index.html), [gpuR](https://cran.r-project.org/web/packages/gpuR/index.html)

With Python:CUDA Python, PyCUDA, PyOpenCL, NumbaPro, Theano etc.