

## Assignment #3

### Bucket Sort in Python

#### Problem Description

Bucket Sort is a naturally partitioning method that well suited for parallel computing. A simple parallel version of bucket sort was introduced in Lecture 6. A working MPI implementation in C was provided (on page 16~18 in Lecture 6 slides) utilizing `MPI_Bcast()`, `MPI_gather()` and `MPI_Gatherv()` methods.

#### Assignment

You are to rewrite the parallel program of Bucket Sort in Python using numpy and mpi4py methods such as `comm.Bcast()`, `comm.Gather()` and `comm.Gatherv()` to sort 64 random integer numbers with 8 processes. Print out the unsorted elements and sorted elements respectively on process 0.

Hint: i) Suppose *unsorted* is your unsorted list of random numbers, you may use `np.logical_and()` method for each process to create its own bucket containing values that fall within its interval in Phase 2 of bucket1 implementation, e.g.,

```
local_bucket = unsorted[np.logical_and(unsorted >= local_min, unsorted < local_max)]
```

ii) Refer to bucket1.c algorithm in Lecture 6 and python sample codes in Lecture 14 if needed.

#### Submission

Submit your source code on Canvas by Nov 6<sup>th</sup>.