Assignment 02: Using scatter reduce and broadcast to perform basic statistics on a data set

Introduction:

In this assignment you will be tasked with performing a statistical analysis of a set of numbers. You are required to calculate the mean and then calculate the standard deviation of the set of numbers. However in this case you will be required to generate all numbers on the first node and using the broadcast and scatter commands you will distribute this data to all nodes that are participating.

Once the numbers have been scattered you will be required to calculate an overall mean for the dataset. However, as there are multiple mean values you must use a reduce command to recieve those values on a coordinator node. It will calculate the overall mean and once calculated will broadcast this back to all nodes for calculating the standard deviation. The results are then reduced on the coordinator node again to compute an overall standard deviation.

Notes:

You have two weeks to do this assignment. Thus the deadline for this assignment will be 2016-04-07 at 23:55. Standard penalties will be applied to work that is submitted so much as a second late. The time of submission as displayed by the moodle will be the reference point for lateness.

You must submit a single zip file (naming does not matter) that contains one c++ file containing your MPI source code. Code that fails to compile will incur a penalty of 30%. The accepted compression formats for your archives are tar.gz/tar.bz2/tar.xz/zip/rar/7z any format outside of this will incur a 10% penalty.

For the purposes of this assignment you will only need four standard header files <iostream>, <cstdlib>, <cmath> and <mpi.h>.

Task List:

01) write a main method that will initialise MPI, figure out the world rank and world size. Rank 0 should be the coordinator while all other ranks should be participants. Then

finalise MPI and return a status of 0 to the OS (5%)

02) write a printArray method that will print out an array to console in a single line. It should accept two parameters a pointer to the array and the size of the array. (2%)

03) write a sum method that takes in a reference to an array and an array size it should return the sum of all the values in that array (3%)

04) write a sumDifferences method that takes in a reference to an array, an array size, and the overall mean of the dataset. It should produce a sum of the square of differences between each value in the dataset and the mean and return this as the result (5%)

05) write coordinator and participant methods that do the following (65%)

- generate the array of numbers (coordinator only). for predicatable results seed the random number generator with the value of 1 and limit their maximum value to 50. (5%)
- determine the size of each partition (coordinator only). Broadcast this to all nodes. (10%)
- scatter the partitions to each node. (10%)
- calculate the mean for this node. Use a reduce operation to gather the overall average. (10%)
- Compute the overall average (coordinator only). (5%)
- Broadcast the overall average to all nodes and then compute the sum of differences (10%)
- reduce the overall sum of differences (10%)
- calculate the standard deviation and print out the dataset, mean and standard deviation (coordinator only) (5%)

06) modify your code to work with any world size and accept a dataset size from the command line. You may assume that the dataset size will be evenly divisible by the world size (10%)

07) Do a comparison of four nodes against a single node on dataset of different sizes. Try to find a crossover point where the four node version is faster than the single node version. Produce a graph containing this cross over point. Provide a short one page commentary on what this graph states about your algorithm (10%)