Homework Assignment 2

General information

The source file allreduce.cpp required for this assignment can be found in the documents section on Minerva. All examples should be compiled and run on the 'Delcatty' cluster (part of the UGent HPC). In order to compile the source file on the interface node of Delcatty, issue:

```
module load ictce
mpicxx -O3 allreduce.cpp -o allreduce
```

To run the example on the interface node

```
mpirun -np 4 ./allreduce
```

where the parameter specified by the –np flag denotes the number of parallel processes that should be started. Note that all processes are started on the interface node itself. For this assignment, this is sufficient. In order to start processes on the worker nodes, you will have to write a job script. I refer to the wiki page on http://hpc.ugent.be for information about how to do this (not required for this assignment).

Modalities

This assignment should be completed **individually**. While it is OK to discuss the solution with fellow students, it is strictly prohibited to exchange and/or copy solutions.

You should **not** write a report for this assignment.

Only hand in the modified **allreduce.cpp** file. **Write your name in this source file as well (in comment).**

The modified allreduce.cpp filename should be renamed to "allreduce_firstname_lastname.cpp". Use the Minerva dropbox to submit your solutions. Do not zip or otherwise pack these the file!

The deadline is Monday, November 3rd 12:00 (noon), CET. This deadline is firm and non-negotiable.

1 Allreduce implementation

Download the C++ source file `allreduce.cpp` from Minerva. Implement the allreduce algorithm in the "allreduceRSAG" routine using only MPI point to point communication routines (MPI_Send, MPI_Recv). Additionally, you might want to use MPI_Comm_rank and MPI_Comm_size.

You must use the "**reduce-scatter – allgather**" algorithm in order to do this. Please refer to the exercise slides for details about this algorithm. Additional instructions can be found in the source file itself.