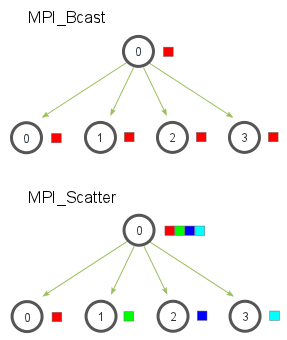
**CS3014 Lab 2: Complex matrix multiplication in MPI**

**Padraic Rowley**

**12304905**

Instead of using senders and receiver’s in MPI to allocate sections of memory to individual multiplication instances within the matrix, I attempted to use the “scatter” and “gather” functions within MPI. Scatter dynamically divides an array into chunks to be dealt with, similar to B\_cast except instead of sending copies of the same process to different places at the same time, it divided the overall array size into “chunks” and dealt with each of these simultaneously.



To divide this correctly I got the total size of the resultant 2D array by multiplying a\_dim2\*b\_dim2 then divided it by an integer “size” which represented the number of processors available. This was the decided chunk size which A would be split into and return its value to C once complete.

After this B was sent over the all the chunks with the same size specifications only using B\_cast so all of B would only be applied to a part of A in each process.

Then the function “teamMatmul” is initialized and after it is carried out with each section af the array dedicated to different sections of the array it shall have the results put back into C using “gather”.

Then finally, the MPI execution enviroment would be closed using “finalize”.

In implementation I managed to get each set of rows and cols in A and B to go to the right places however when finding a way to implement the function specifically for each separate multiplication instance it proved more complicated than expected and any results I managed to get were either extremely wrong, failed to run at all, or did not manage to run the processes completely.