

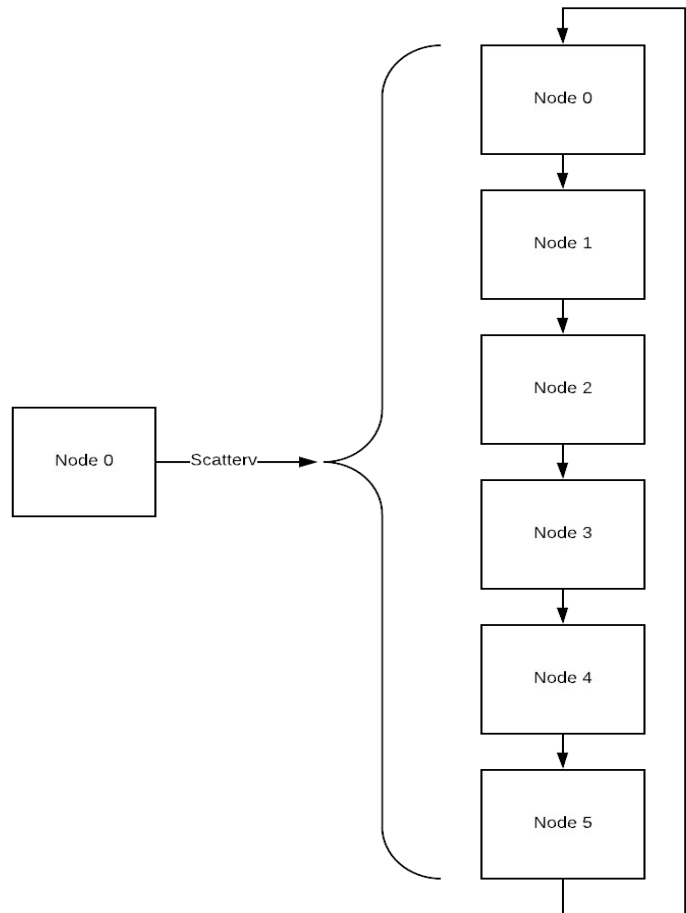
Assignment PA2

For this assignment I used a round-robin distribution process as the basis for distributing data between the processes. In practice it was not actually implemented this way, but the results are indistinguishable from if it was. The following diagram intends to illustrate the round-robin process, where one item is given to the first node in the system, the one to the next, and so on, after looping back to the beginning.

In the actual code this was implemented by testing conditions with the division remainder of the number of items divided by the number of nodes. This code was:

```
((int) (N / numranks)) + ((N % numranks > rank) ? 1 : 0)
```

The number of items per node was found and then rounded down by casting it to an integer, the second part of the expression then uses a ternary operator, for simplicity, to add 1 if the current rank is less than the remainder. What this last part serves to do is check if that rank should hold an extra piece of overflow to catch all the data. This is the same as the last round in the round-robin process where data is only distributed to some of the nodes since it is not evenly distributable.



As far as timing is concerned, I did not observe the difference between using MPI_Scatter and MPI_Scatterv were significantly different. The timing data, which can be seen in the PA2 Data.ods file, shows the time to scatter as being extremely inconsistent even with a decent sized dataset, furthermore,

as is shown in the following graph, the values for each of the functions were within margin of error. If there were any differences I would have expected MPI_Scatterv to be slower since the data is being scattered more inconsistently.

