What if the array cannot be broken into equal pieces [for a scatter]?

You pad the array so that it can be broken into same-sized pieces.

So, with MPI_Scatter, does the 'boss' processor get a part of the task sent to it?

Yes.

I'm a bit confused with how one might use MPI_Reduce. How would one, say, do a reduction with an array of 12 elements into a single value? Where do you put the array?

Each processor sums into a PartialSum value and then all the PartialSum values get returned and summed into the GlobalSum.

So PartialSum is always a single value?

 Doesn't have to be. That’s why there is a #elements argument there. For example, it you were reducing an X array and a Y array at the same time, the PartialSum could be a 2-element array.

How long has MPI been in existence in this form?

Just checked WikiPedia. Work started in the summer of 1991. For more information, see: https://en.wikipedia.org/wiki/Message_Passing_Interface

Looks like a pretty interesting history.

How do MPI_Barriers work? Can you assign a barrier as 'Barrier A' or is it based on each processor hitting that line in the code?

My understanding is that it is doing a count of how many processors have called MPI_BARRIER(). As each calls MPI_BARRIER(), it freezes. Once the count reaches the number of processors, they all un-freeze.

It [Project #7] has to be in the dgx right?

The “DGX” is the name of the GPU cluster. You will use the CPU cluster, which you get at through the submit-* machines, just like you did when you used the DGX.

We can make the tags whatever is useful to us, correct?

Yes – just be sure that sending and receiving use the same tags.

The optimal processor number [Compute:Communicate notes] will grow with data size, right?

Correct.