Postpone Intro to HPC for now

1 History of MPI

- 1980 - 1995
  1. every vendor had their own message passing libraries
  2. ORNL created PVM (parallel virtual machine); it was free; used master/slave programming model
  3. Europe message passing library standard PAMACS; wanted a world standard, in 1995 led to MPI; MPI uses the SPMD programming paradigm, i.e., a single program runs on all processors
- 1988 - 1992: intense research effort to automatically parallelize serial programs

2 Overview of MPI

A MPI communicator defines a group of processes to be used when running an MPI program. MPI has a predefined communicator named `mpi_comm_world` which consists of a single group of processes: $1, 2, ..., p-1$. Each process is assigned a rank. MPI groups can be restricted to subgroups and one can define *intercommunicators* and *intracommunicators* between subgroups. Most of the class, we will only use `mpi_comm_world`.

All MPI programs must contain the following.

1. Call `mpi_init(ierror)`: initializes the MPI environment (`ierror` is an int)
2. Call `mpi_comm_size(comm, p, ierror)`: p then is the number of MPI processes
3. Call `mpi_comm_rank(comm, rank, ierror)`: rank of the executing processor
4. Call `mpi_finalize(ierror)`: clean up environment