

MPI

- MPI_Init (&argc, &argv)
- MPI_Finalize()
- MPI_Comm_size (MPI_COMM_WORLD, &nproc); - # processes in -world.
- MPI_Comm_rank (MPI_COMM_WORLD, &rank); - id of the process (0 = master)

blocking

- MPI_Send (void *buffer, int count, MPI_Type, int dest, int tag, MPI_Comm world)
- MPI_Recv (buf, count, MPI_Type, int src, int tag, MPI_Comm world, MPI_Status *status)
- MPI_Sendrecv (send, recv, MPI_Comm world, MPI_Status *status)

Types: CHAR, SHORT, INT, LONG, UNSIGNED_CHAR, UNSIGNED_SHORT, UNSIGNED_LONG, FLOAT, DOUBLE, LONG_DOUBLE, BYTE, PACKED

asynchronous

- MPI_Isend (buf, count, type, dest, tag, comm, MPI_Request *request)
- MPI_Irecv (buf, count, type, src, tag, comm, MPI_Request *request)
- MPI_Test (MPI_Request *request, int *flag, MPI_Status *status)
- MPI_Get_count (MPI_Status status, type, int *count) \uparrow MPI_SUCCESS

blocking

- MPI_Wait (MPI_Request *request, MPI_Status *status)
 - \rightarrow status \rightarrow MPI_SOURCE
 - status \rightarrow MPI_TAG
- MPI_Probe (int source, int tag, MPI_Comm world, MPI_Status *status) - looks for a message
 - \uparrow MPI_ANY_SOURCE
 - \uparrow MPI_ANY_TAG

- MPI_Bcast (void *buffer, int count, MPI_Type, int root, MPI_Comm comm)
 - \uparrow (source - all other processes receive)
- MPI_Reduce (void *sendbuf, void *recvbuf, count, type, MPI_Op op, int root, MPI_Comm comm)
 - \uparrow (who will get the result)
 - MPI_SUM, PROD, MAX, MIN,
 - MAXLOC, MINLOC, LAND, LOR,
 - XOR, BAND, BOR, BXOR