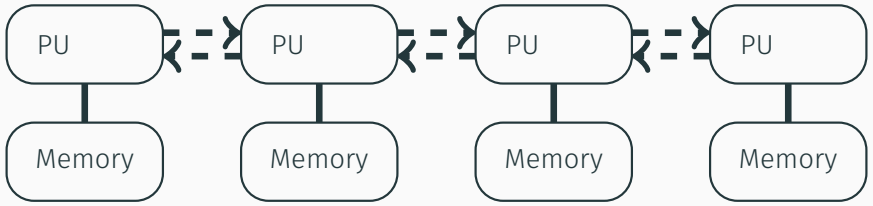
Introduction to MPI

Jeremy Iverson

College of Saint Benedict & Saint John's University

background



- A standard for explicit distributed memory parallel computation.
- Many implementations available, both open-source and proprietary.

execution model

- Uses the *SPMD* model of parallelism
  - All processes execute the same program.
    - Different processes carry out different actions by conditional execution of code based on processes' rank.
    - Processes can communicate with each other by sending explicit messages

library api

- Requires inclusion of `mpi.h` header file.
- `MPI_Init()`
- `MPI_Finalize()`
- `MPI_Comm_size()`
- `MPI_Comm_rank()`
- `MPI_Send()`
- `MPI_Recv()`
- MPI uses *communicators* to organize processes. Processes can only communicate with other processes in the same communicator. The base communicator to which all processes belong is called `MPI_COMM_WORLD`.
- Programs can deadlock due to improperly ordered or unmatched point-to-point communications.