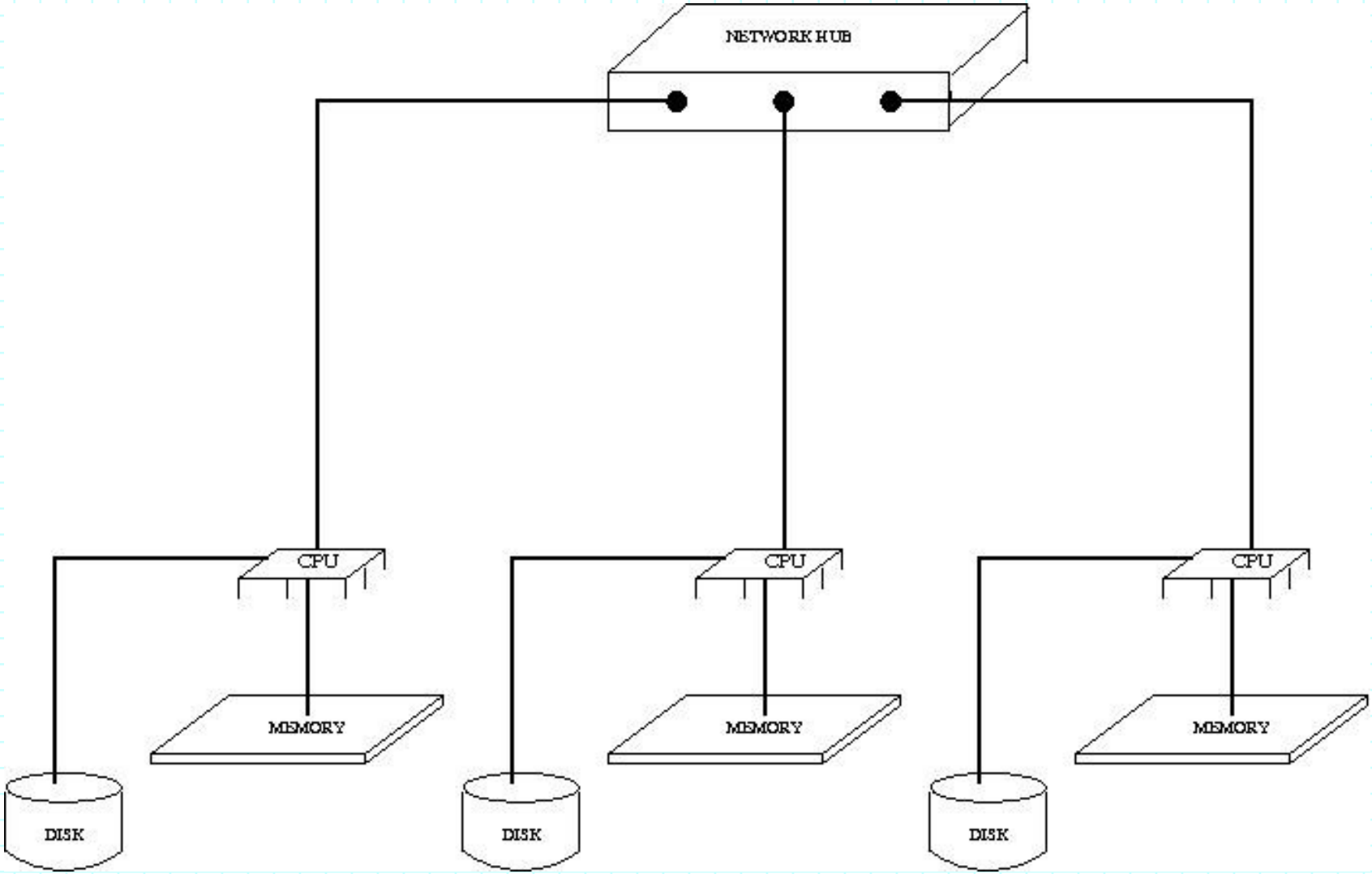
High Performance Computing Systems

- Distributed Memory
- MPI

Doug Shook via Roger Chamberlain

Distributed Memory



Distributed Memory

- Network is critical
- Communication plays a large role in performance
  - Data distribution
  - Latency

- Do not confuse this with distributed computing

Message Passing

- Design considerations

- Many different systems available
  - CORBA
  - SOAP
  - D-Bus
  - DCOM
  - MPI

Message Passing Interface

- Formed around 1991
  - Language independent protocol
- Works with processes
  - Commonly referred to as processors among programmers
- Based upon eight basic concepts

Communicators / Groups

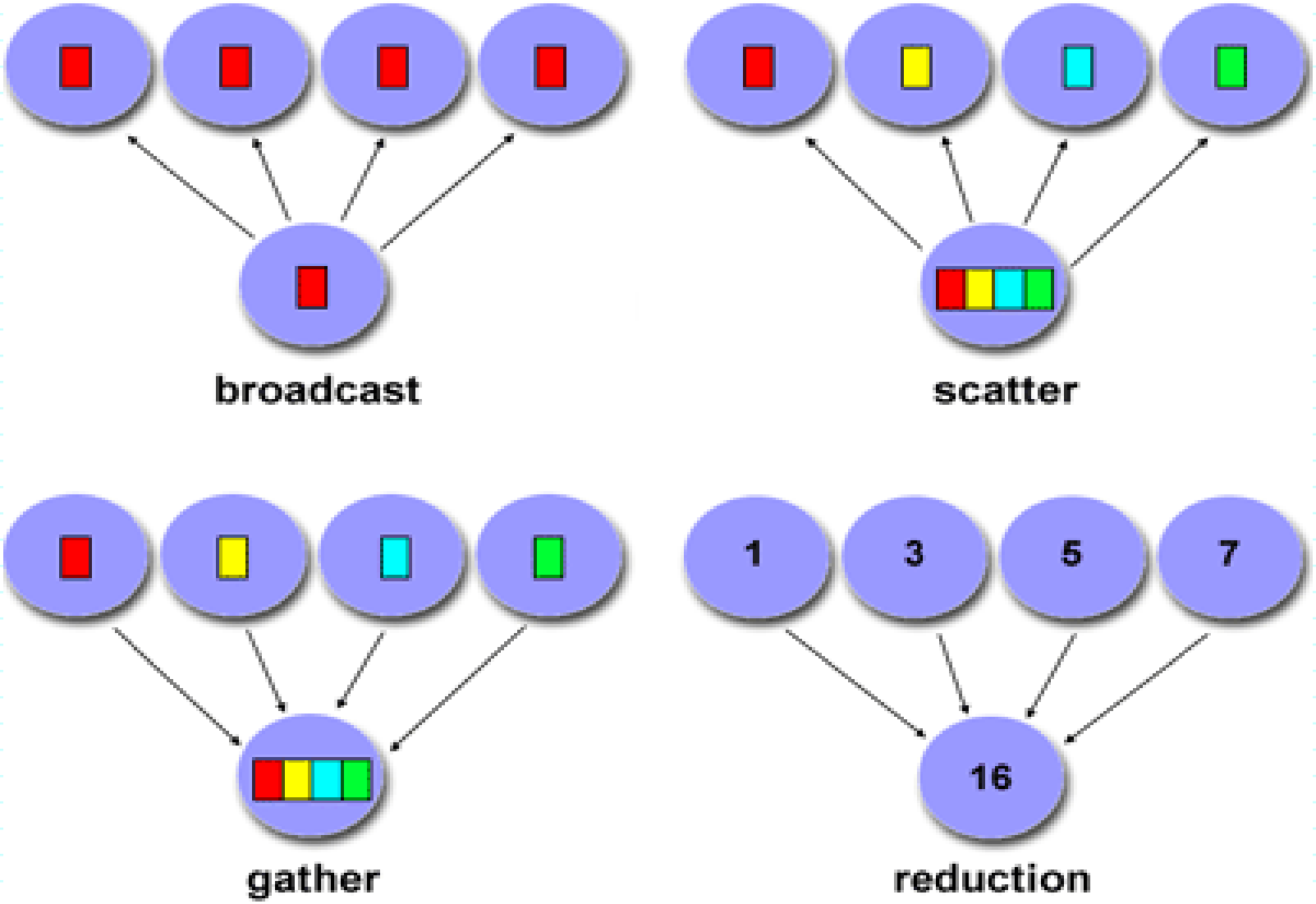
- Processes are broken down into groups
- Communicators encompass group(s)
- For some tasks, a single group / communicator is plenty

Point-to-Point

- Basic building blocks for process communication
- Useful for patterned or irregular communication
- Specification allows for both blocking and non-blocking methods

Collective

- Communication among all processes within a group



Data Types

- Functions pass variables, not defined types
  - MPI_INT, MPI_CHAR, MPI_DOUBLE, etc.
- Can also define contiguous types

Message Passing Details

- For communication to succeed:
  - Sender must specify a valid destination (rank)
  - Receiver must specify a valid source (rank)
  - The communicator must be the same
  - Tag must be the same
  - Receiver's buffer must be large enough

MPI vs. OpenMP

MPI	OpenMP
Designed for distributed memory systems (processes)	Designed for shared memory systems (threads)
Requires more programming – harder to debug	Directives can be added gradually
Performance dependent on communication network	Performance dependent on CPU architecture

OpenMPI

- Merger of three MPI implementations

- Provides functions based on the MPI specification

OpenMPI - Setup

- Available in repos of most Linux distros

- #include <mpi.h>
- MPI_Init, MPI_Finalize
  - Called by all MPI processes
- mpicc
- mpiexec <path to executable>

OpenMPI – Practical considerations

- Each process is given an ID (rank)
- For work that only needs to be done once, use ID to specify "main" process to do that work
- Breaking up the work depends on the number of processes
- MPI_COMM_WORLD

OpenMPI – Practical considerations

- Performing a reduction:
  - Each process keeps its own local value
  - "main" process reduces them with MPI_Reduce:
    - Address of buffer to be sent (variable)
    - Address of receiving buffer (variable)
    - Number of elements sent
    - Datatype of elements (MPI)
    - Reduction operation (MPI)
    - Rank of main process
    - Communicator

OpenMPI – Performance Monitoring

- MPI_Wtime – timing function
  - Can be used in all processes to monitor individual performance
  - Or only in main process to monitor total performance
  - Resolution can be gotten with MPI_Wtick

Hello World

Blocking vs. Non-blocking

- MPI_Send, MPI_Recv
- MPI_Isend, MPI_IRecv

Contiguous Data Types