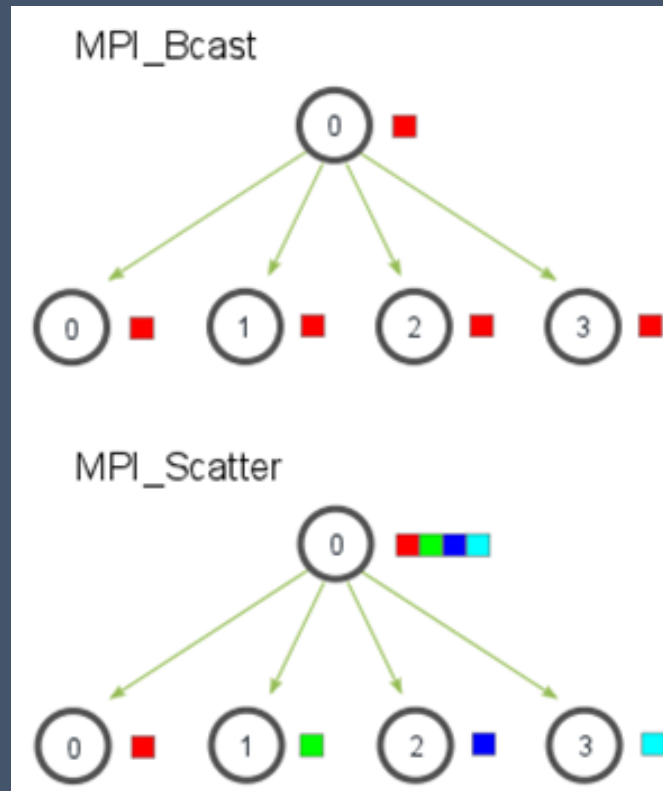


Message Passing Interface (MPI)

- Standard for message passing between processes in high performance computing (HPC) applications.
- Parallel distributed computing model.
- Processes are executed in parallel
 - As separated processes in the same host.
 - As separated processes in different hosts.
- Primitives for point-to-point and multicast communication
 - send/recv
 - broadcast
 - scatter
 - gather

Message Passing Interface (MPI)

- Broadcast (bcast)

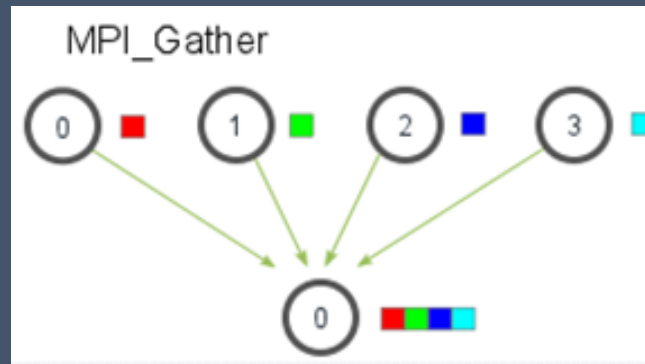


Source: mpitutorials.com

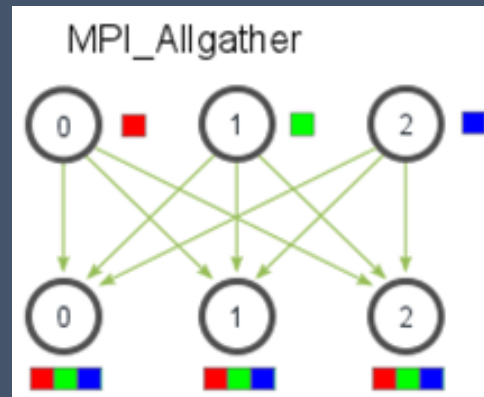
- Scatter

Message Passing Interface (MPI)

- Gather



- Allgather



Source: mpitutorials.com

Message Passing Interface (MPI)

- Mac/Linux: Install OpenMPI (current version 4.1.5)
- Windows: Install Microsoft MPI
 - <https://docs.microsoft.com/en-us/message-passing-interface/microsoft-mpi>
 - Set C:\Program Files\Microsoft MPI\Bin to PATH
- Python package mpi4py
- mpirun (mpiexec) -h hostfile -n N python script.py
- Hostfile
 - Tells MPI where to run processes
 - Syntax <hostname> <number_of_processors>

Exercises

DS_Examples/mpi/mpi_example.py

1. Change the example so that each task includes a command to be executed by the worker
 - ADD if worker adds all parameters
 - SUB if worker subtracts the last two parameters from the first one
 - POW if worker should sum the first two parameters and elevate it to the power given by the third one

The worker will then execute the operation and send the result back, along with its rank. If the operation is not known, it will return the string 'UNKNOWN'

Exercises

2. Change the example so the root node can cope with any number of processors
 - If there are fewer data than processors, fill the rest with None
 - If there are more data than processors, distribute the data among the processors

Hint: use lists