Matrix Vector Multiplication

Week 4

What we have covered so far…

- Intro
- Hardware
- Parallelism
- Amdahl’s Law
- Gustafson’s Law
- Dependence

What we have covered so far… 2

- OMP
  - Parallel for
  - Reduce sum
- MPI
  - Send / Receive
  - Reduce
  - Broadcast

Code we have seen

- Hello world
- Array Sum
- Trapezoidal Rule

Class activity

$$\begin{bmatrix} 2 & 5 & 2 \\ 1 & 0 & -2 \\ 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} -2 & 1 & 0 \\ -2 & 2 & 1 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} \quad & \quad & \quad \\ \quad & \quad & \quad \\ \quad & \quad & \quad \end{bmatrix}$$

Write a sequential program for Matrix vector multiplication. You must take input from user or read from file. Use dynamic memory allocation.

Questions to think about.

- How can you parallelize the program? What parts are dependent? Proof!
- How can you distribute the task into smaller tasks? (also draw diagram)
- What will you need to communicate among different processes?
- How will you join back the solutions of smaller tasks?
- Using Amdahl's Law can your parallel solution run faster with more than M processor where M is the number of rows of the matrix. Can you have a strategy that is optimal for MxN processors? (N is number of Columns)

Write an MPI parallel version of your matrix vector multiplication.

Last Activity

- Matrix vector multiplication
- Multiply each row in different process with the vector.
- Problem – No speedup with more N processors
  - N = number of rows

Today – Write a MPI program for Matrix Vector Multiplication

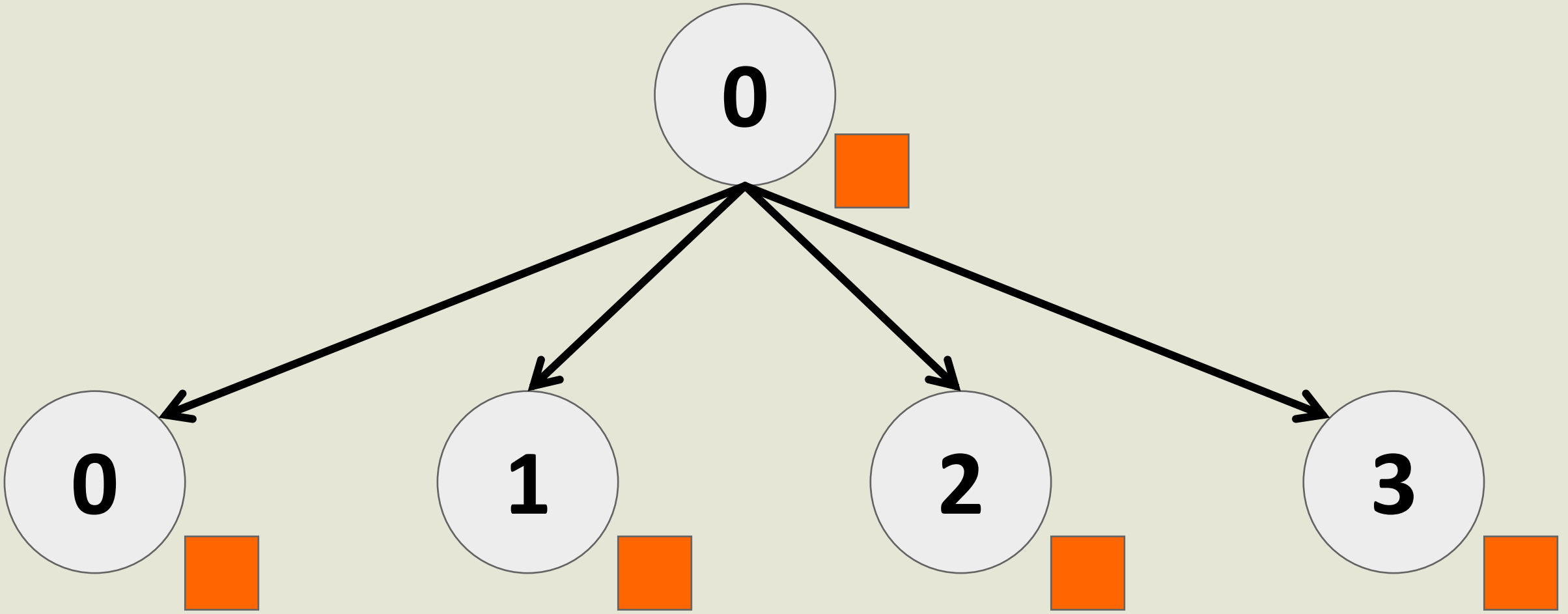
- What task each process must perform?
- How to split the tasks?
- What needs to be communicated?
- What (data) needs to joined/collected after process have completed their task.

Today – Write a MPI program for Matrix Vector Multiplication

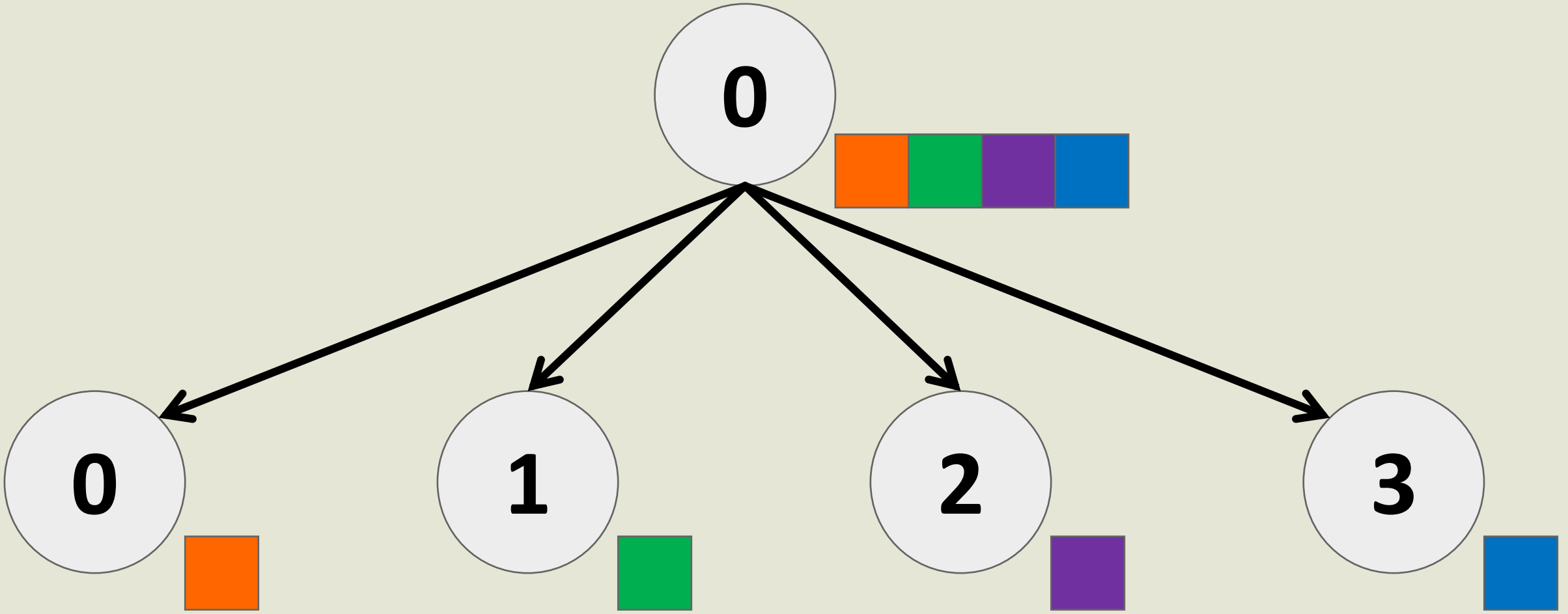
- Use of MPI_Send
- Use of MPI_Receive
- Use of MPI_Bcast

Is it easy?

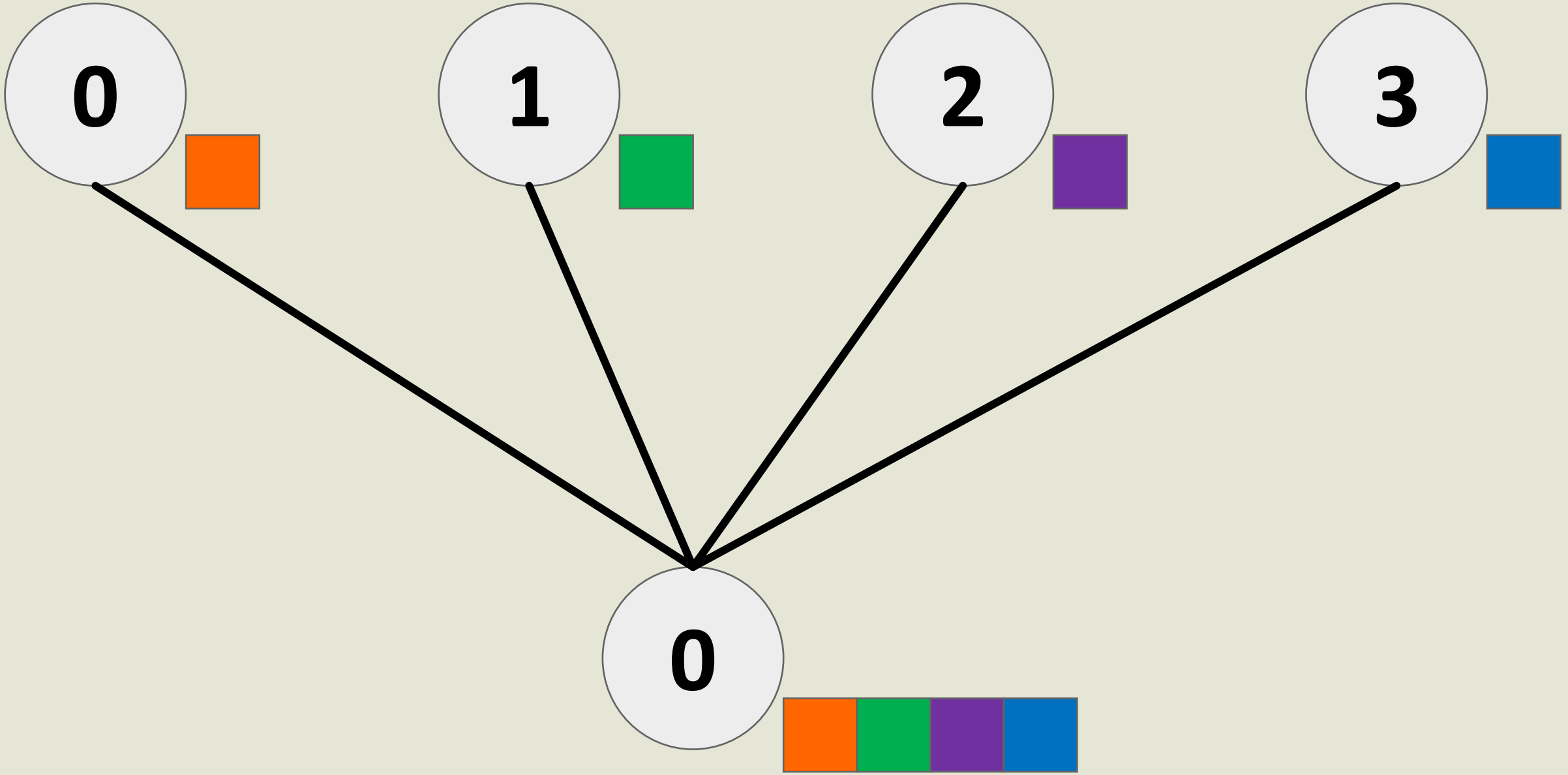
MPI_Bcast



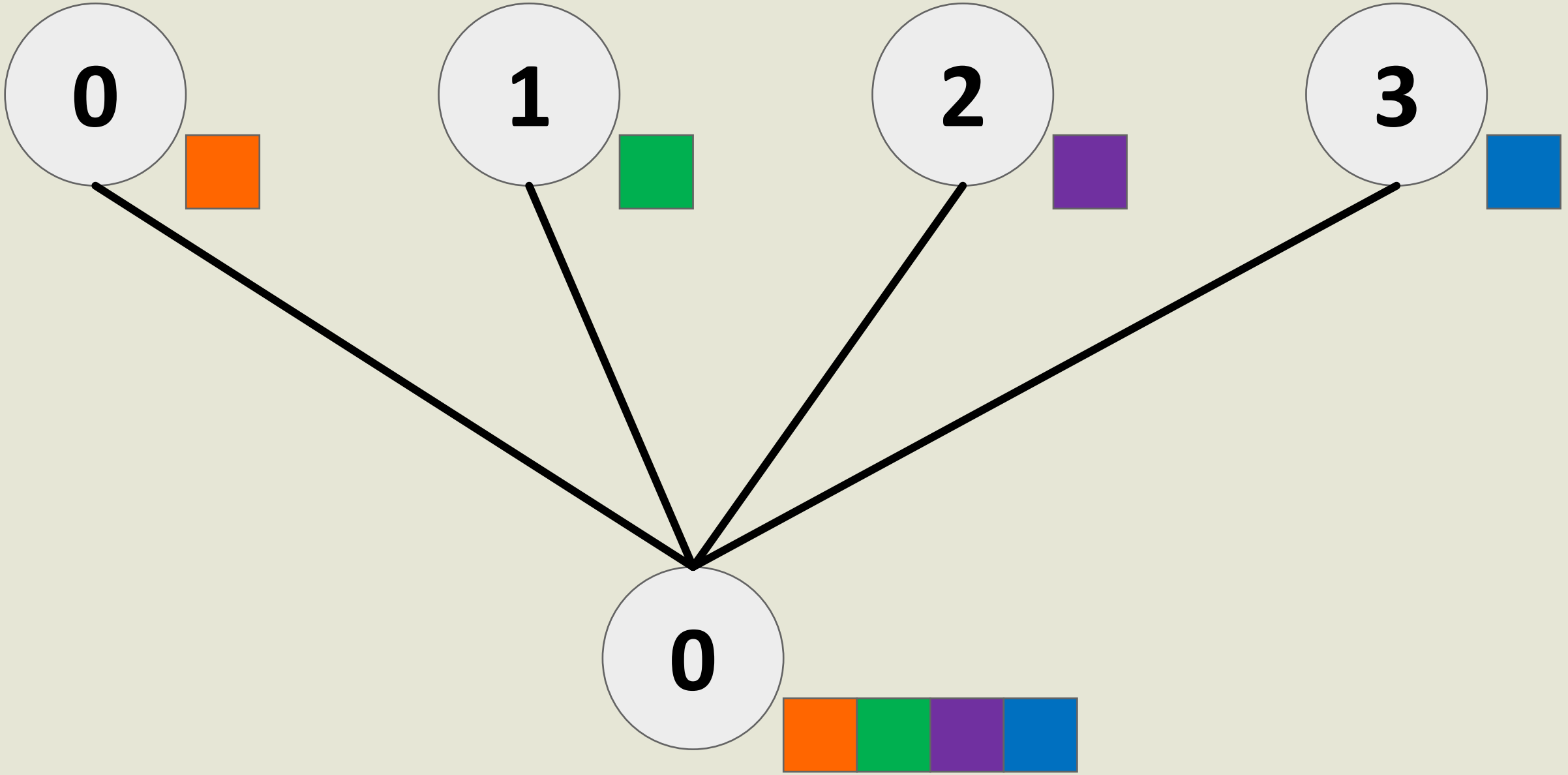
MPI_Scatter



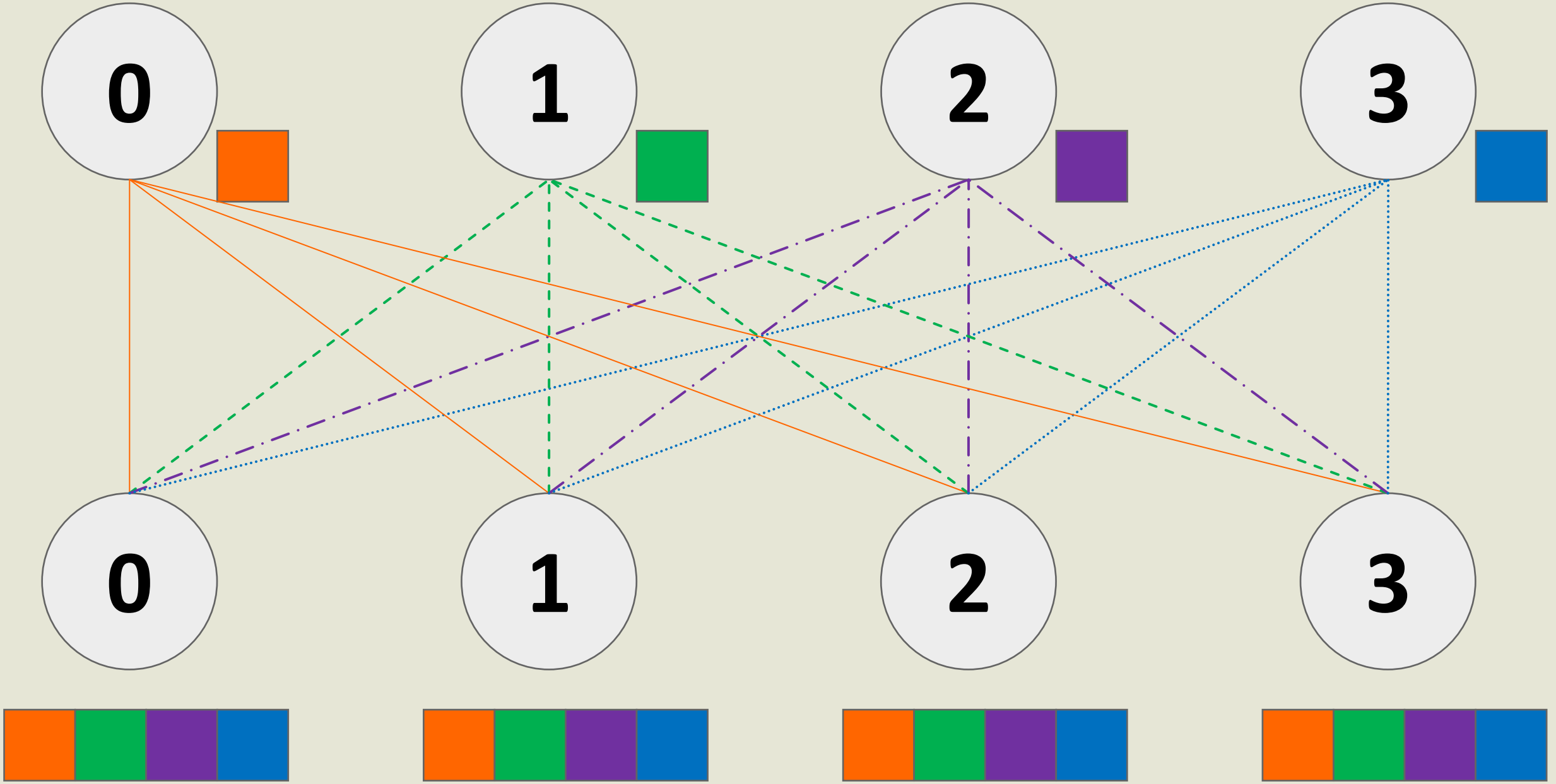
MPI_Gather



MPI_Gather



MPI_Allgather



```
int MPI_Scatter(
void          *sendbuf,
int           sendcount,
MPI_Datatype  sendtype,

void          *recvbuf,
Int           recvcount,
MPI_Datatype  recvtype,

int           root,
MPI_Comm      comm
);
```

MPI_Scatter

```
int MPI_Gather(
void          *sendbuf,
int           sendcount,
MPI_Datatype  sendtype,

void          *recvbuf,
int           recvcount,
MPI_Datatype  recvtype,

int           root,
MPI_Comm      comm
);
```

MPI_Gather

```
int MPI_Allgather(
void          *sendbuf,
int           sendcount,
MPI_Datatype  sendtype,

void          *recvbuf,
int           recvcount,
MPI_Datatype  recvtype,

MPI_Comm      comm
);
```

MPI_Allgather

QUIZ on Wednesday!

Quiz

All MPI functions studied up till now!

Their parameters and how they behave on root and slave process.

End of Lecture