15-440
Distributed Systems
Recitation 10

Tamim Jabban
Project 3

• Using Message Passing Interface (MPI) to apply the K-Means algorithm

• Due date: November 12th
  • You should be working on the parallel versions now! 😊
Agenda

• Today, we’ll be re-implementing the Parallel Sum program from last week

• We’ll use collective routines to do so
Collective Communication

• Collective communication allows you to exchange data among a group of processes

• It must involve all processes in the scope of a communicator

• The communicator argument in a collective communication routine should specify which processes are involved in the communication

• Hence, it is the programmer's responsibility to ensure that all processes within a communicator participate in any collective operation
Patterns of Collective Communication

- There are several patterns of collective communication:
  1. Broadcast
  2. Scatter
  3. Gather
  4. Allgather
  5. Alltoall
  6. Reduce
  7. Allreduce
  8. Scan
  9. Reducescatter
Patterns of Collective Communication

- There are several patterns of collective communication:
  
  1. Broadcast
  2. Scatter
  3. Gather
  4. Allgather
  5. Alltoall
  6. Reduce
  7. Allreduce
  8. Scan
  9. Reducescatter
Scatter and Gather

• Scatter distributes distinct messages from a single source task to each task in the group
• Gather gathers distinct messages from each task in the group to a single destination task

```c
int MPI_Scatter ( void *sendbuf, int sendcnt, MPI_Datatype sendtype, void *recvbuf, int recvcnt,
                    MPI_Datatype recvtype, int root, MPI_Comm comm )
```

```c
int MPI_Gather ( void *sendbuf, int sendcnt, MPI_Datatype sendtype, void *recvbuf, int recvcount,
                    MPI_Datatype recvtype, int root, MPI_Comm comm )
```
Reduce and All Reduce

- **Reduce** applies a reduction operation on all tasks in the group and places the result in one task.
- **Allreduce** applies a reduction operation and places the result in all tasks in the group. This is equivalent to an MPI_Reduce followed by an MPI_Bcast.

```c
int MPI_Reduce ( void *sendbuf, void *recvbuf, int count, MPI_Datatype datatype, MPI_Op op, int root, MPI_Comm comm )
```

```c
int MPI_Allreduce ( void *sendbuf, void *recvbuf, int count, MPI_Datatype datatype, MPI_Op op, MPI_Comm comm )
```