

Informe comparación `my_MPI_Bcast` VS `MPI_Bcast`

Daniel Yanac

15 de abril de 2015

1. `MPI_Bcast`

`MPI_Bcast` es uno de los estándares de técnicas de comunicación colectiva. Durante un broadcast, el maestro envía la misma data a todos los esclavos que estén en el mismo comunicador.

```
MPI_Bcast(void* data, int count, MPI_Datatype datatype,  
int root, MPIComm communicator);
```

2. *`my_MPI_Bcast`*

La idea es poder simular la misma funcionalidad interna de `MPI_Bcast` pero usando solo `MPI_Recv` y `MPI_Send`, tomando los mismos parametros que se le pasa a `MPI_Bcast`.

```
void my_MPI_Bcast(void* data, int count, MPI_Datatype  
datatype, int root, MPIComm communicator)  
{  
    int rank, size;  
    MPI_Comm_rank(communicator, &rank);  
    MPI_Comm_size(communicator, &size);  
    if (rank == root)  
    {  
        int i;
```

```

        for(i = 0; i < size; i++) {
            if (i != rank)
                MPI_Send(data, count, datatype, i, 0, communicator);
        }
    else
    {
        MPI_Recv(data, count, datatype, root, 0, communicator,
        MPI_STATUS_IGNORE);
    }
}

```

3. Comparación

Datos - Procesadores	Tiempo <i>my-MPI_Bcast</i>	Tiempo <i>MPI_Bcast</i>	Diferencia
10 - 2	0.0	0.0	0.0
1000 - 2	0.0007	0.0	0.0007
100000 - 2	0.0007	0.0002	0.0005
10 - 4	0.0	0.0	0.0
1000 - 4	0.0009	0.0	0.0009
100000 - 4	0.0007	0.0002	0.0005
10 - 8	0.0001	0.0	0.0001
1000 - 8	0.0013	0.0001	0.0012
100000 - 8	0.0044	0.0007	0.0037
10 - 16	0.0016	0.0001	0.0015
1000 - 16	0.0018	0.0002	0.0016
100000 - 16	0.0096	0.0016	0.008
10 - 32	0.0218	0.0002	0.0216
1000 - 32	0.0221	0.0003	0.0218
100000 - 32	0.0370	0.0076	0.0294

4. Conclusiones

Se puede observar que la implementación de *my-MPI_Bcast* es menos eficiente a la implementa de *MPI_Bcast* propia de *MPI*. Esto quizás se deba

a que la implementación interna de *MPI_Bcast* utilice otras funciones además de las ya mencionadas `MPI_Recv` y `MPI_Send` para hacer el paso de mensajes más eficiente.