Parallel Programming

Prof. **Paolo Bientinesi**

`pauldj@aices.rwth-aachen.de`

WS 17/18

HPAC

RWTH AACHEN
UNIVERSITY

Collective Communication

```
            Barrier
Broadcast   ↔   Reduce
  Scatter   ↔   Gather
Allgather   ↔   Reduce-scatter
           Allreduce
            Alltoall
               ⋮
```

References

- "Collective Communication: Theory, Practice, and Experience", Chan, Heimlich, Purkayastha, van de Geijn.
  (FLAME working note #22)
- Collective Communications in MPI

  `http://www.mcs.anl.gov/research/projects/mpi/tutorial/gropp/node72.html`

Collective Communication

- Synchronization
  `Barrier` ← **Almost never needed!**
- Data Movement
  `Broadcast, Scatter, Gather, Allgather, Alltoall`
- Reductions
  `Reduce, Reduce-scatter, Allreduce, Scan, ...`

All processes invoke the same operation with the same arguments. No tags.

Collectives are **blocking**, but **not synchronous**: The routines return as soon as their participation in the communication is complete. No guarantee about the status of the receiving processes.

`int MPI_BCast(...)`

Before:

Node$_0$	Node$_1$	Node$_2$	Node$_3$
		α	

After:

Node$_0$	Node$_1$	Node$_2$	Node$_3$
α	α	α	α

- How would you implement the broadcast in terms of individual sends and receives?
- How many steps does it take to broadcast to np processes?