Communication algorithms

- Reduction
- All-to-all
Types of reduction

- Scalar reduction: every process contributes one number
  - Perform some commutative associate operation
- Vector reduction: every process contributes an array of numbers
Parallelizing reduction
Parallelizing reduction

- Naive algorithm: every process sends to the root
Parallelizing reduction

- Naive algorithm: every process sends to the root
- Spanning tree: organize processes in a k-ary tree
Parallelizing reduction

- Naive algorithm: every process sends to the root
- Spanning tree: organize processes in a k-ary tree
- Start at leaves and send to parents
- Intermediate nodes wait to receive data from all their children
Parallelizing reduction

- Naive algorithm: every process sends to the root
- Spanning tree: organize processes in a k-ary tree
- Start at leaves and send to parents
- Intermediate nodes wait to receive data from all their children
- Number of phases: $\log_k p$
**All-to-all**

- Each process sends a distinct message to every other process
- Naive algorithm: every process sends the data pair-wise to all other processes

```
<table>
<thead>
<tr>
<th>Input Data</th>
<th>MPI_Alltoall Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0 0 1 2 3</td>
<td>P0 0 4 8 12</td>
</tr>
<tr>
<td>P1 4 5 6 7</td>
<td>P1 1 5 9 13</td>
</tr>
<tr>
<td>P2 8 9 10 11</td>
<td>P2 2 6 10 14</td>
</tr>
<tr>
<td>P3 12 13 14 15</td>
<td>P3 3 7 11 15</td>
</tr>
</tbody>
</table>
```

https://www.codeproject.com/Articles/896437/A-Gentle-Introduction-to-the-Message-Passing-Inter
Virtual topology: 2D mesh

- Phase 1: every process sends to its row neighbors
- Phase 2: every process sends to column neighbors
Virtual topology: hypercube

- Hypercube is an n-dimensional analog of a square (n=2) and cube (n=3)
- Special case of k-ary d-dimensional mesh

<table>
<thead>
<tr>
<th>#Dim</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
</table>

https://en.wikipedia.org/wiki/Hypercube