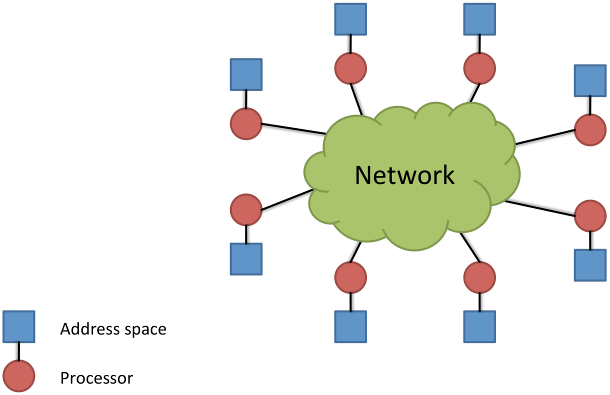
Project 5:
Parallel programming with Message Passing Interface (MPI)

Pratyuksh Bansal, Olaf Schenk

Università della Svizzera italiana

November 3, 2021

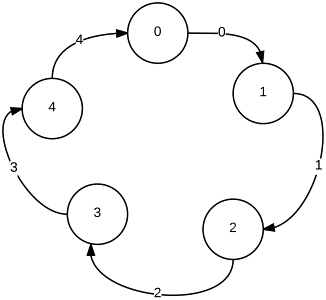
Message-passing model



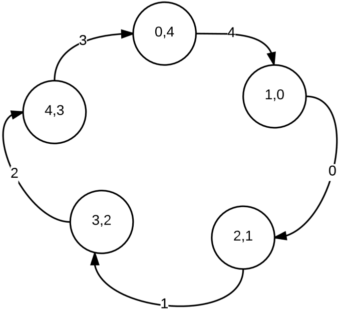
Overview of exercise sheet

1. Ring addition using MPI.
2. Ghost cells exchange between neighboring processes.
3. Parallelizing the Mandelbrot set using MPI.
4. **Option A** : Parallel matrix-vector multiplication and the Power method.
5. **Option B** : Parallel PageRank Algorithm and the Power method.

Ring addition using MPI



Ring addition using MPI



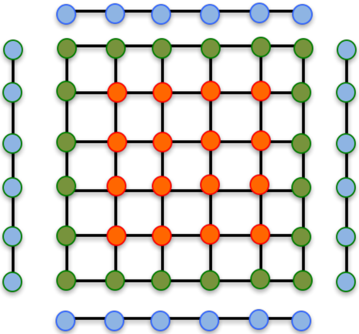
Ghost cells exchange

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Figure: 4×4 Cartesian topology.

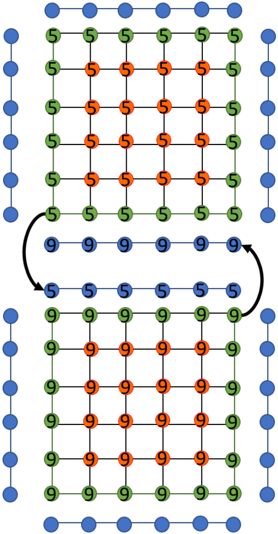
Question: How to create a Cartesian topology?

Ghost cells exchange

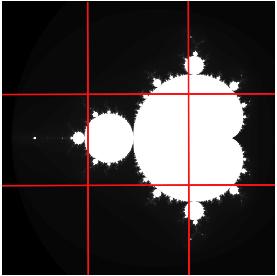


- Boundary cells need the information in ghost cells for some computation.
- The data in ghost cells depends on neighboring processes.

Ghost cells exchange



Parallelizing the Mandelbrot set using MPI



- Similar exercise performed with OpenMP.
- Create partition; that is create a Cartesian topology.
- Define the physical domain for each processor, then compute.
- Send local data to the root processor.

A : Parallel matrix-vector multiplication & the Power method

- A be a $n \times n$ matrix.
- Compute largest eigenvalue/eigenvector of A?
  Use power method.

Algorithm 1 Power method

x is random vector of length n.
for $i = 1$ to N **do**
    $x \leftarrow x/||x||$
    $x \leftarrow Ax$
end for
$\lambda_{\text{max}} = ||x||$
$v_{\text{max}} = x/\lambda_{\text{max}}$

A : Parallel matrix-vector multiplication & the Power method

Given : Matrix dim n, number of processors p.
Assumption : n is divisible by p.

Step 1: Generate matrix A

- Each processor generates its own rows.

Step 2: Matrix-vector multiplication

Step 2: Implement power method

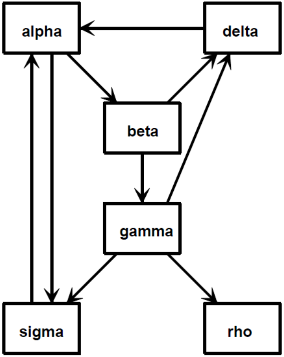
Experiments: Strong scaling and Weak scaling.

B : Parallel PageRank Algorithm & the Power method

- Used in the initial version of Google search engine.
- Ranks all the web pages.
- How? Generate the transition matrix A, then solve

$$x = Ax, \tag{1}$$

  x is the vector of page ranks.
- Solve (1) with Power method!



B : Parallel PageRank Algorithm & the Power method

Transition matrix A?

$$A = pGD + ez^{\top},$$

- G is a sparse matrix,
- D is a diagonal matrix,
- e and z are vectors and
- probability $0 \leq p \leq 1$.

Algorithm 2 PageRank

```
G ← pGD
Compute z.
x_i = 1/n.
for i = 1 to N do
    x ← Gx + e(z · x)
end for
```

B : Parallel PageRank Algorithm & the Power method

Given : Serial implementation.

To Do :

- Implement changes with MPI to get a parallel version.
- Benchmark your code on the provided datasets.
- Analyze and describe your results.

Questions?