The MPI Message-passing Standard (VI)
Practical use - II

SPD Course

16/04/2010

Massimo Coppola

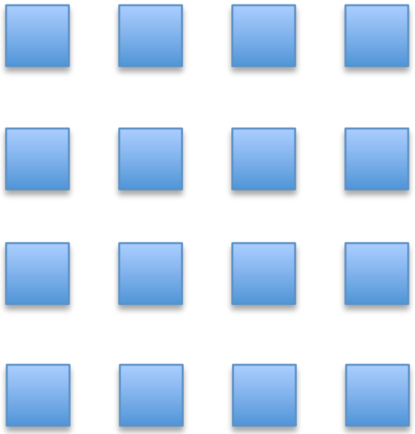
Previously done :

- Define a program with 2 processes
  - they send back and fort a data buffer, the second process executes an operation on the data (e.g. sum 1).
  - Verify after a given number N of iterations, that the expected result is achieved.

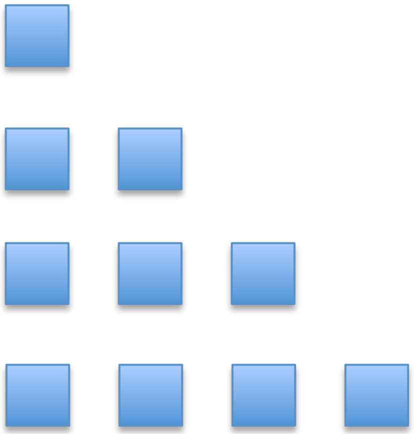
Exercises

- Define a datatype for a square matrix, with parametric size. Define a datatype for its lower triagular matrix. Define one for its upper triangular.

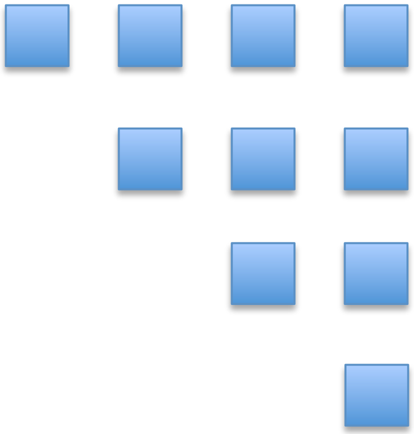
$A_{i,j}$ *i,j* in *1.. n*



$A_{i,j}$ *i≥j*

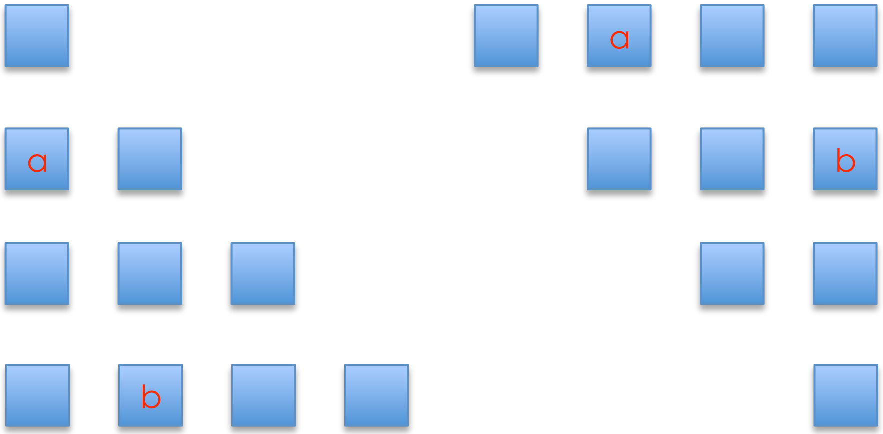


$A_{i,j}$ *i≤j*



Exercises

- In the two-process program, initialize randomly a square matrix, send the lower triangular and receive it back as upper triangular in the same buffer.
  - Is the result a symmetric matrix?
  - Do you need to modify one of the two triangular datatypes?
- In the end we want $A_{i,j} = B_{j,i}$



Exercises

- How do you implement an asynchronous communication with given asincrony?
  - Implement a communication with asynchrony 1
  - Implement a communication with asynchrony K

- Assigned asynchrony of degree K: asynchronous communication (sender does not block) which becomes synchronous if more than K messages are still pending.
- Receiver can skip at most K receives before sender blocks

- Can you rely on MPI buffering?
- How would you implement a fixed size buffer?