COSC 6374
Parallel Computation

Dense Matrix Operations

Edgar Gabriel
Spring 2009

Terminology

- Dense Matrix: all elements of the matrix contain relevant values
  - Typically stored as 2-D array, (e.g. `double a[16][32];`
- Sparse matrix: most elements of the matrix are zero
  - Optimized storage techniques
    - Band matrices: store only the relevant diagonales of the matrix
    - Highly irregular sparse matrices: store the coordinates of every non-zero element together with the content
    - Boeing-Harwell format: exploit certain regularities (e.g. nearly constant number of entries per row or column)
    - Jagged Diagonal storage format: see Boing Harwell format

Replication vs. Communication

- Large data items typically distributed across multiple processes
  - What is large?
- Small data items can replicated on all processes or communicated whenever required
  - Costs for communication: ~ network latency
  - Costs for replication: ~ memory consumption + ~ repeated computation operations

Matrix operations: $B = c \times A$

- Multiplying a Matrix `A` with a constant `c`
- Constant `c` is definitely small and is thus replicated on all processes
  - E.g. compiled in the code
  - Read from a configuration file
- Operation does not require any communication to be performed
  - Trivially parallel
- Operation can be performed independent of the way the matrix has been distributed across the processes

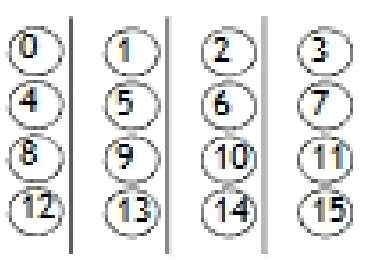
Matrix Operations: $B = A^T$

- Transpose a Matrix
  - Often not necessary, since the operations (e.g. Matrix-vector multiply) can be (easily) reformulated for Matrix-Transpose-vector multiply operations and avoid the data transpose
  - Operations requiring the transpose: multi-dimensional FFT
- Assumption:
  - Matrices `A, B` are square
  - Element `A[x][y]` should by on the same process as element `B[x][y]`

    -> requires communication across the processes

$B = A^T$: One element per process

- Initial data distribution: one element of the Matrix A per process



- Process with coordinates (x,y) needs to send its data item to the process with the coordinates (y,x) and receive its data item from (y,x)