THE REGIONAL

WEATHER FORECASTING SYSTEM SKIRON

Parallel Implementation of The Eta Model

L. Boukas, N. Mimikou

N. Missirlis

Department of Informatics

University of Athens

(visit: http://skiron.di.uoa.gr)

G. Kallos

Department of Applied Physics

University of Athens

(visit: http://forecast.uoa.gr)

Contents of the Presentation

- Description of SKIRON
- Application of Domain Decomposition Techniques
- Data Mapping
- Communication
- Performance

SKIRON: A Regional Weather Forecasting System

- Has been developed for operational use at the Hellenic National Meteorological Service for regional weather forecasts in the Mediterranean Region
- Is a reliable and computationally efficient system which produces forecasts, particularly useful for predicting local atmospheric conditions
- At present the SKIRON system is run daily at the HNMS on an HP Exemplar SPP-1600 with 16 processors
- The initial and boundary condition fields are received from ECMWF. The boundaries are updated every three hours

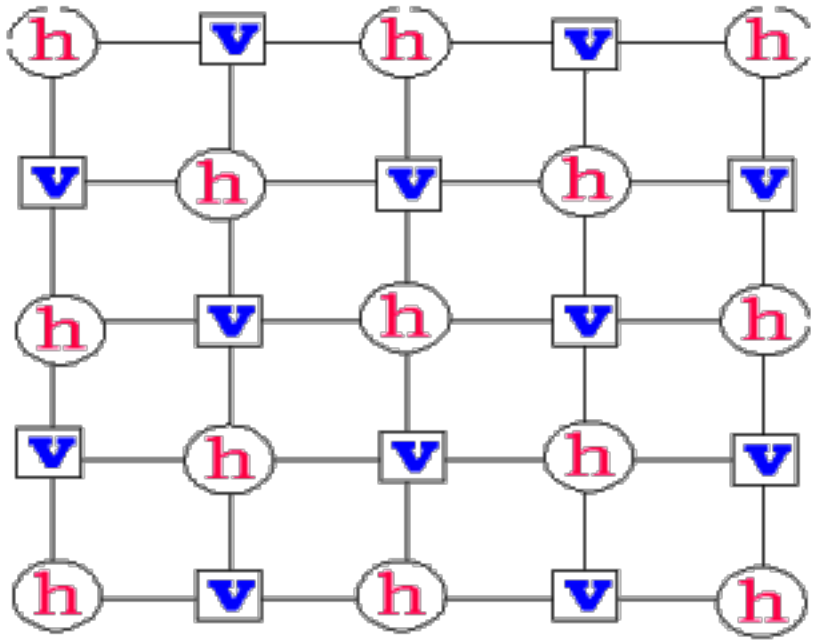
SKIRON: A Regional Weather Forecasting System

- It consists of several modules:
  - Pre-processing
  - The Parallel Eta model
  - Post-processing
- In the Pre-processing phase the system can use
  - The ECMWF analysis and forecasts' gridded data
  - The NCEP analysis and forecasts' gridded data
  - Any other global gridded model output, or output from 3D or 4D, data assimilation system such as LAPS (NOAA/FSL)
- The Post-processing modules are several. One part is the graphical representation of the results

The Eta Model

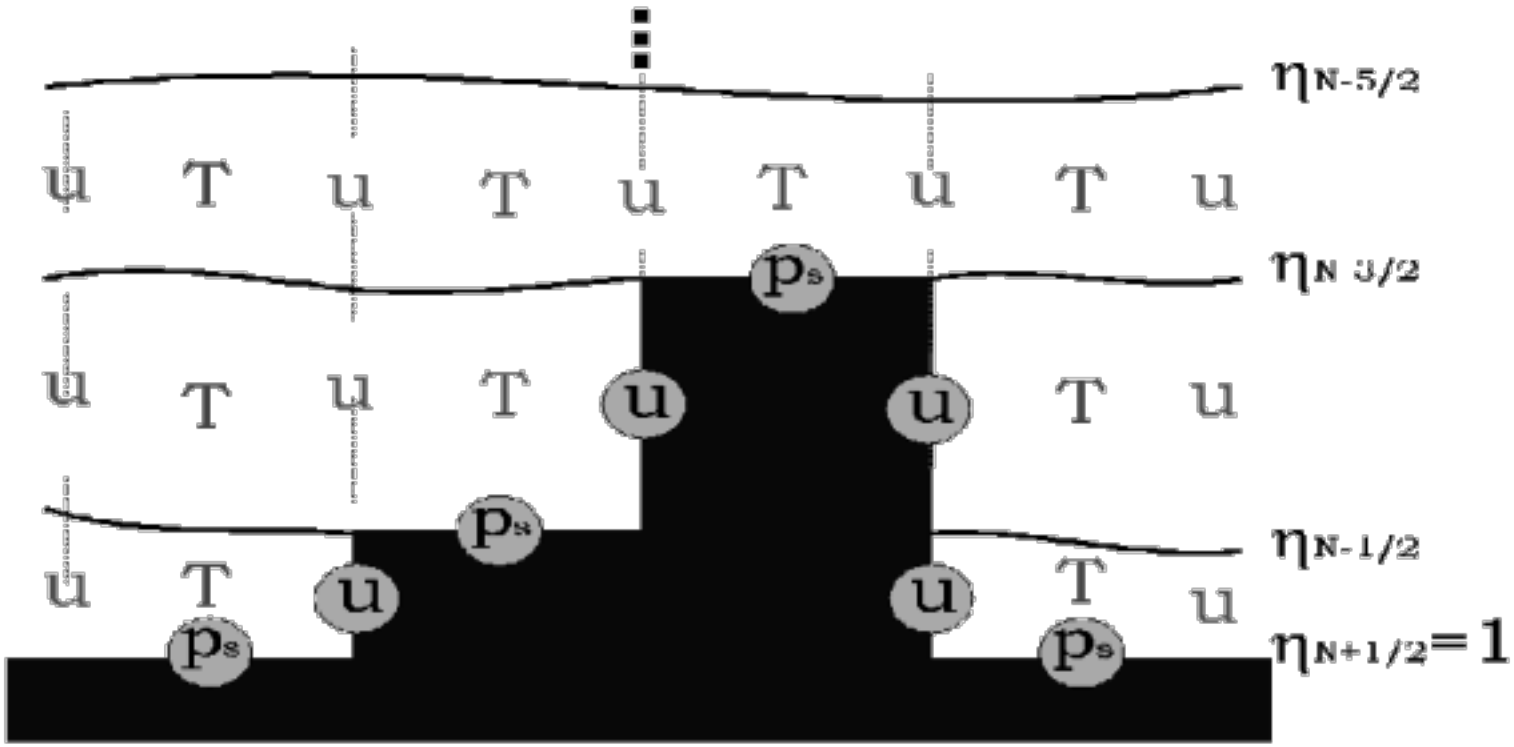
- **Developed by** NCEP (and Belgrade University). Originally optimized for vector based architectures, then transformed into the two dimensional version.
- **Is a** Limited area-Regional scale atmospheric model
- **Uses:**
  - The eta vertical coordinate
  - an Arakawa E grid
  - the Janjic horizontal advection scheme
- **Model Physics:** The physical part of the Eta model is based on several sophisticated parameterization schemes.
- **Model Dynamics:** Designed as a hydrostatic model, uses the primitive equations based on the hydrostatic approximation
- **Discretization with** finite differences
- **Explicit** Horizontal numerical scheme
- **Implicit** Vertical numerical scheme

Horizontal grid structure



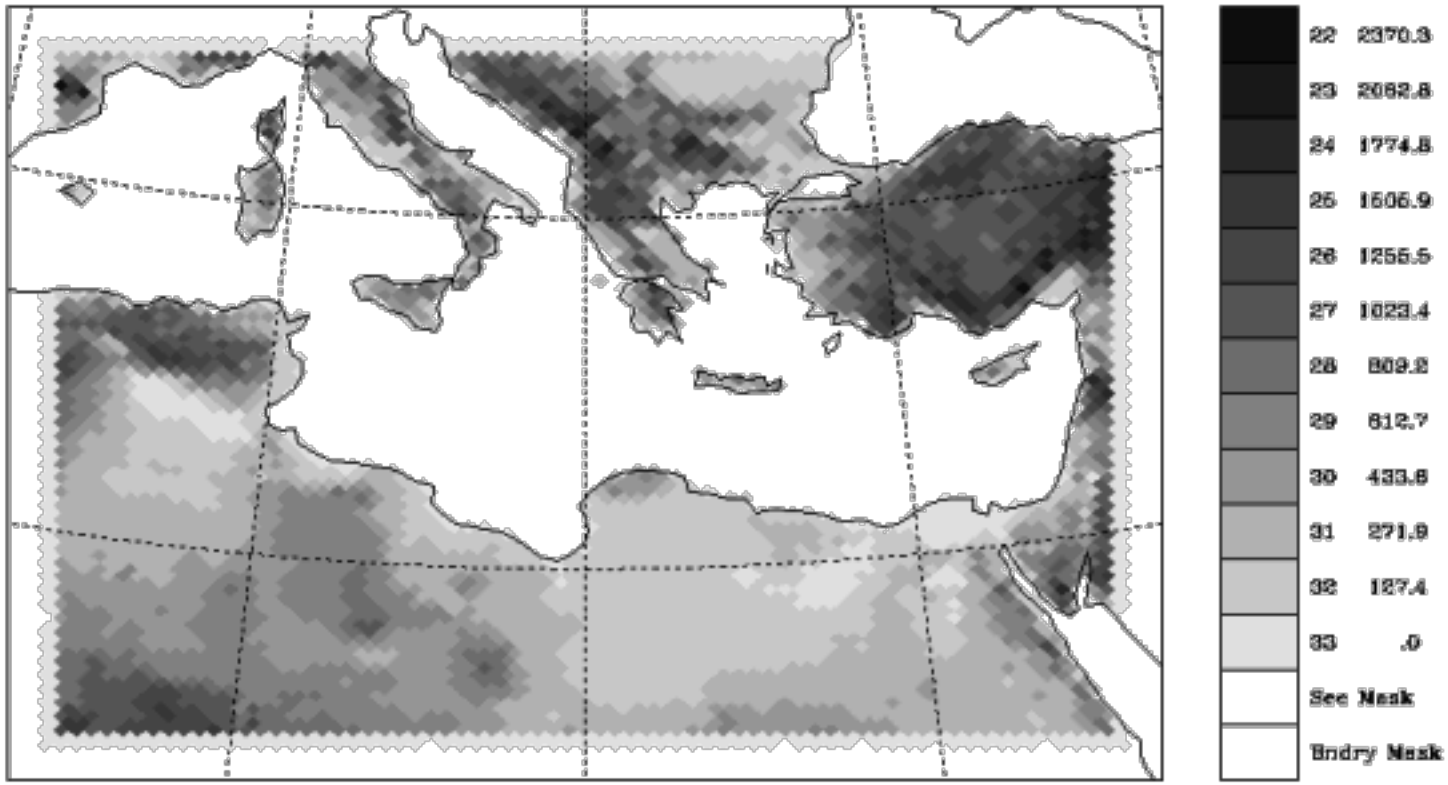
In the horizontal mesh, the Eta model is defined over the semi-staggered E grid (good performance in simulating smaller-scale processes)

Vertical grid structure



The mountains in the Eta system are represented as grid-box mountain blocks.

Model mountain representation

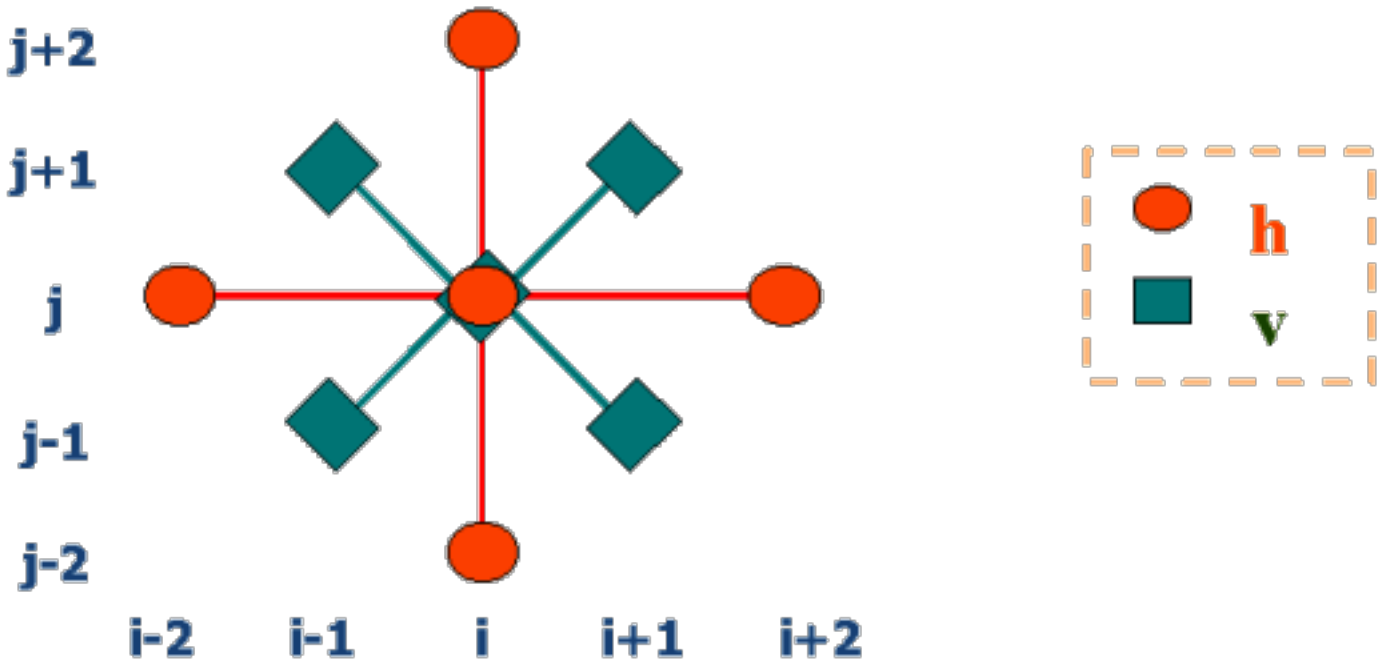


Horizontal distribution of the eta mountains inside the model domain

THE ETA MODEL ROUTINE FORECASTING SYSTEM

- Meteorological parameters (geopotential, wind components and humidity) are collected from an external data source.
- the acquitted data are decoded and interpolated into the Eta model grid structure. Surface parameters are specified in the model grid.
- the model is executed over the specified forecasting period
- the forecast data are further processed in order to prepare the input for the graphical presentation

Stencil for the horizontal discretization



In every grid point the stencil for the horizontal discretization is applied

Domain Decomposition
Principles

- Decompose the original domain into subdomains
- Assign each subdomain to a different processor
- Each processor solves the problem using the original code
- There is only a single code to be maintained for both sequential and parallel computations
- The sequential code is reused entirely in the parallelization
- To keep the computations consistent with the sequential code inter-processor communication is needed

Domain Decomposition
Applied on Eta

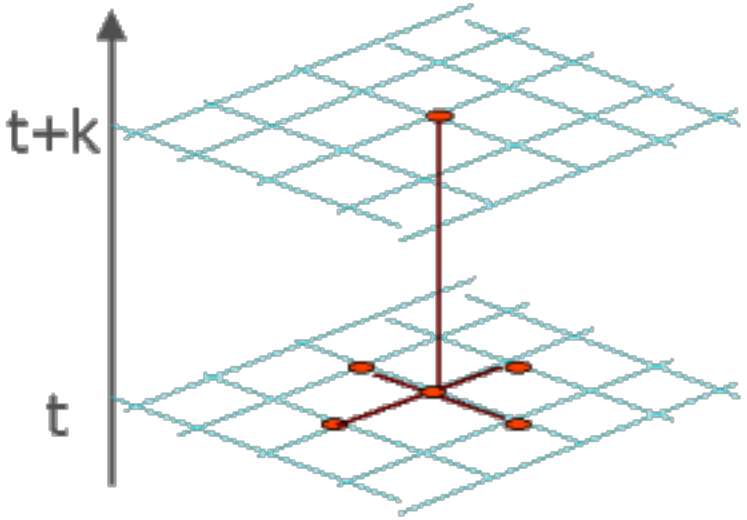
- Question: Can Domain Decomposition be applied on the Eta computations ?
- Answer: Yes !
- **How ?**
  - The computations in the horizontal mesh use explicit finite differences
  - This means that each grid point in the next time level is computed using only values from grid points in the previous time level
  - So, the computations for each grid point in the advanced time level are independent and can be carried out in parallel by assigning a group of these points to each of the available processors
  - The computations in the vertical direction use implicit schemes which are difficult to parallelize. We didn't parallelize these computations

Numerical Schemes

Computation of a grid point in the horizontal mesh using **explicit** numerical schemes

Next time level

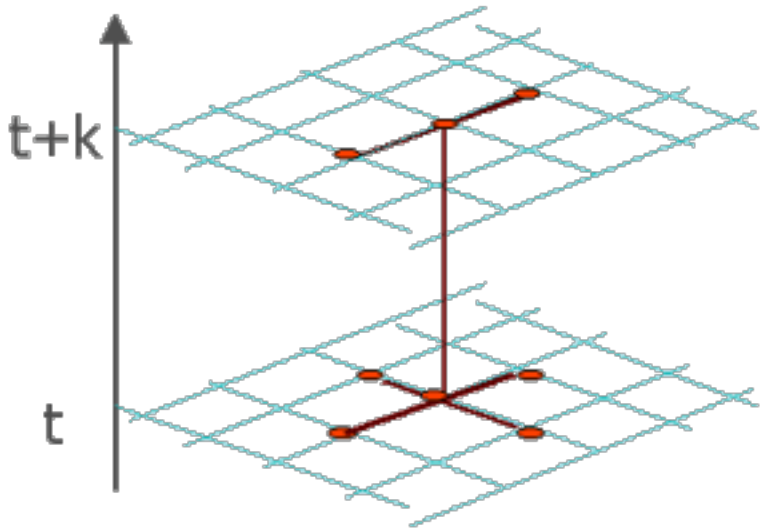
Previous time level



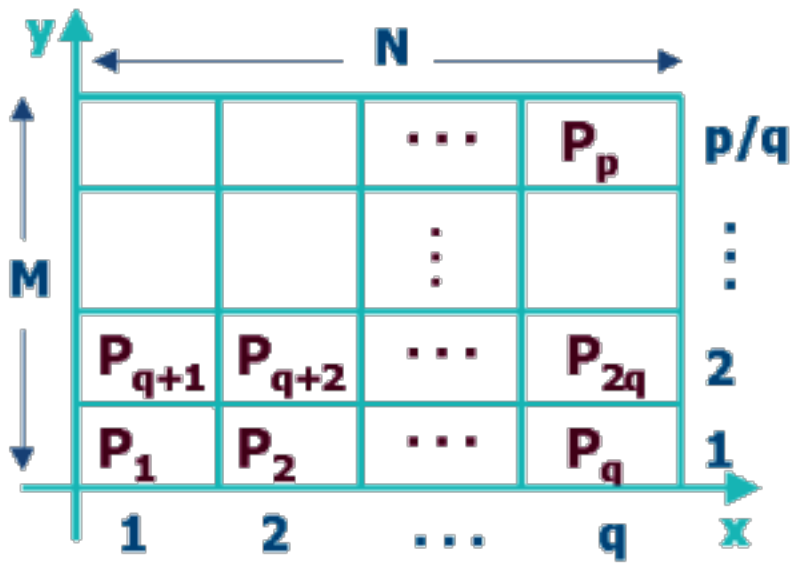
Computation of a grid point in the horizontal mesh using **implicit** numerical schemes

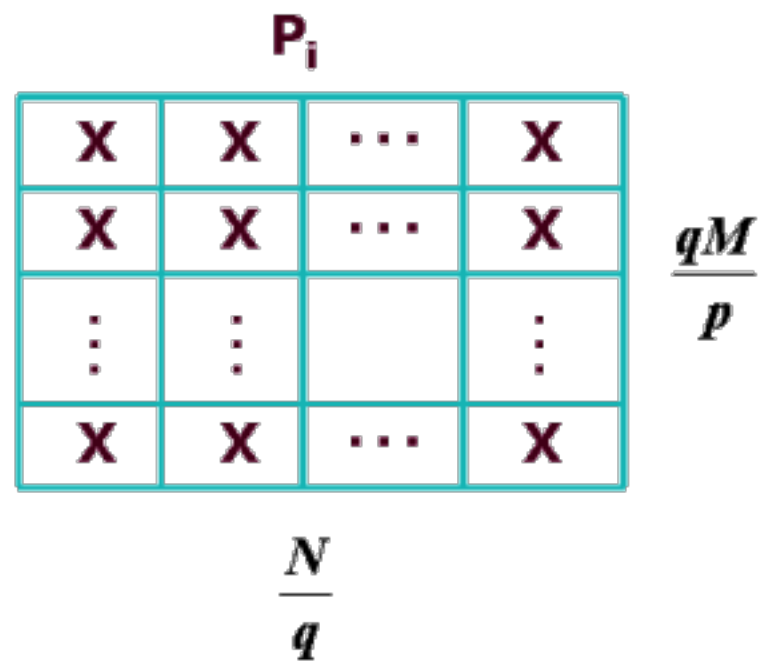
Next time level

Previous time level



Domain Decomposition & Mapping





'X' denotes a grid point

Determination of Subdomain

Problem

Determine the subdomain such that

$$r = \min \frac{\text{Communication Time}}{\text{Computation Time}}$$

Solution

In case there is no need for load balancing it is best to assign to each processor a square subdomain of size equal to $\sqrt{\frac{NM}{p}}$

a square side = $\sqrt{\frac{NM}{p}}$

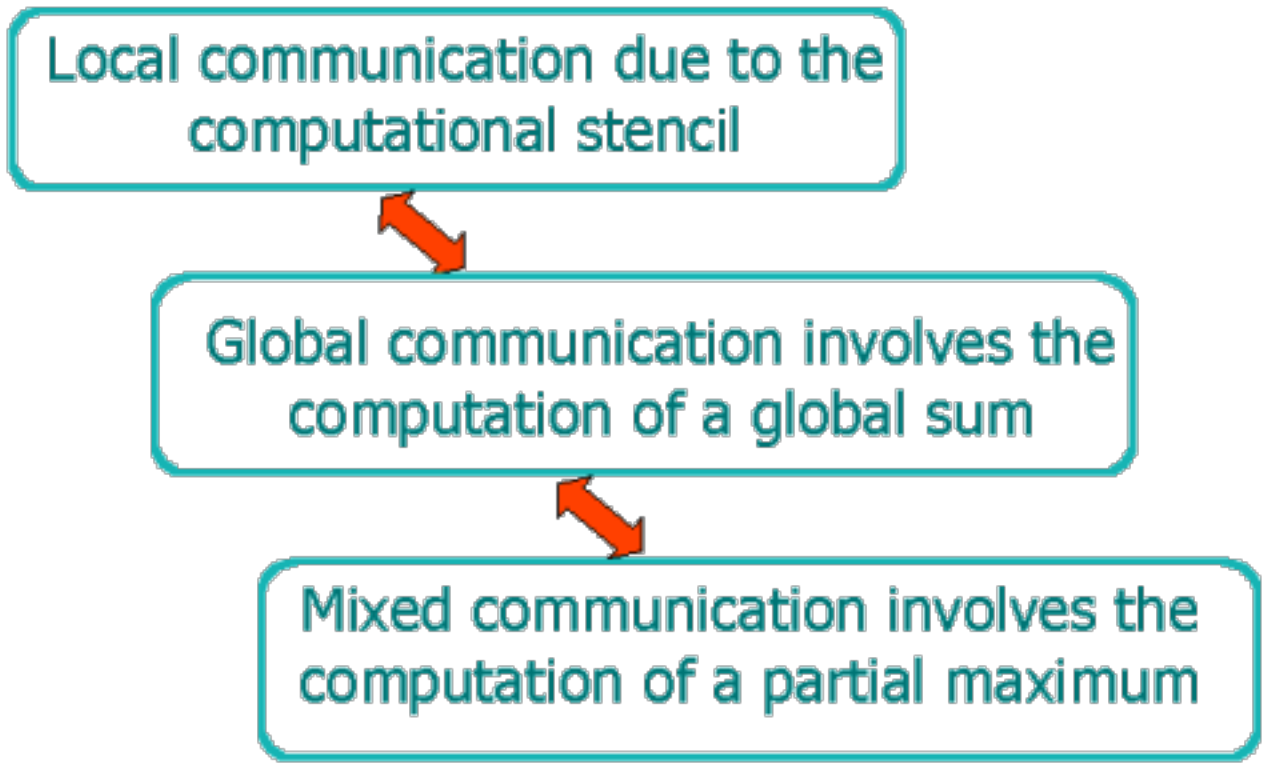
$$\textit{If} \quad \frac{t_s}{t_w} < \textit{side of square}$$

$$\textit{then} \qquad r = O(\sqrt{p})$$

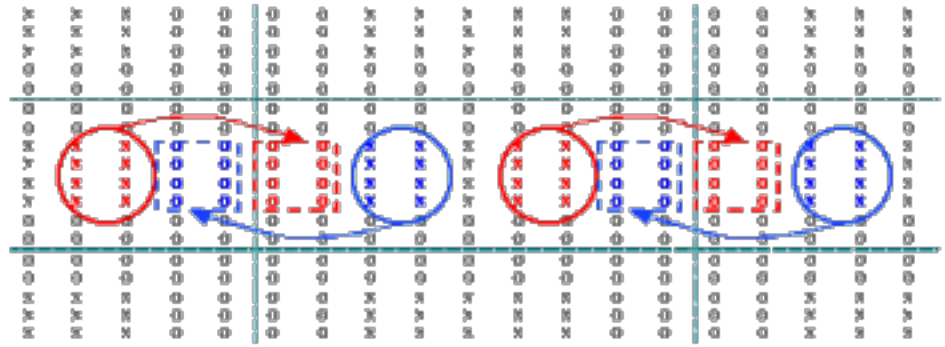
$$\textit{otherwise} \qquad r = O(p)$$

Types of communication

- Local communication
- Global communication
- Mixed communication

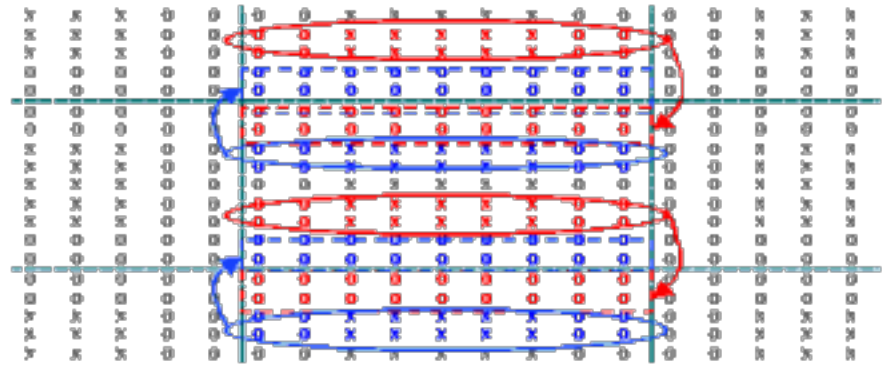


Local communication



Column Exchange

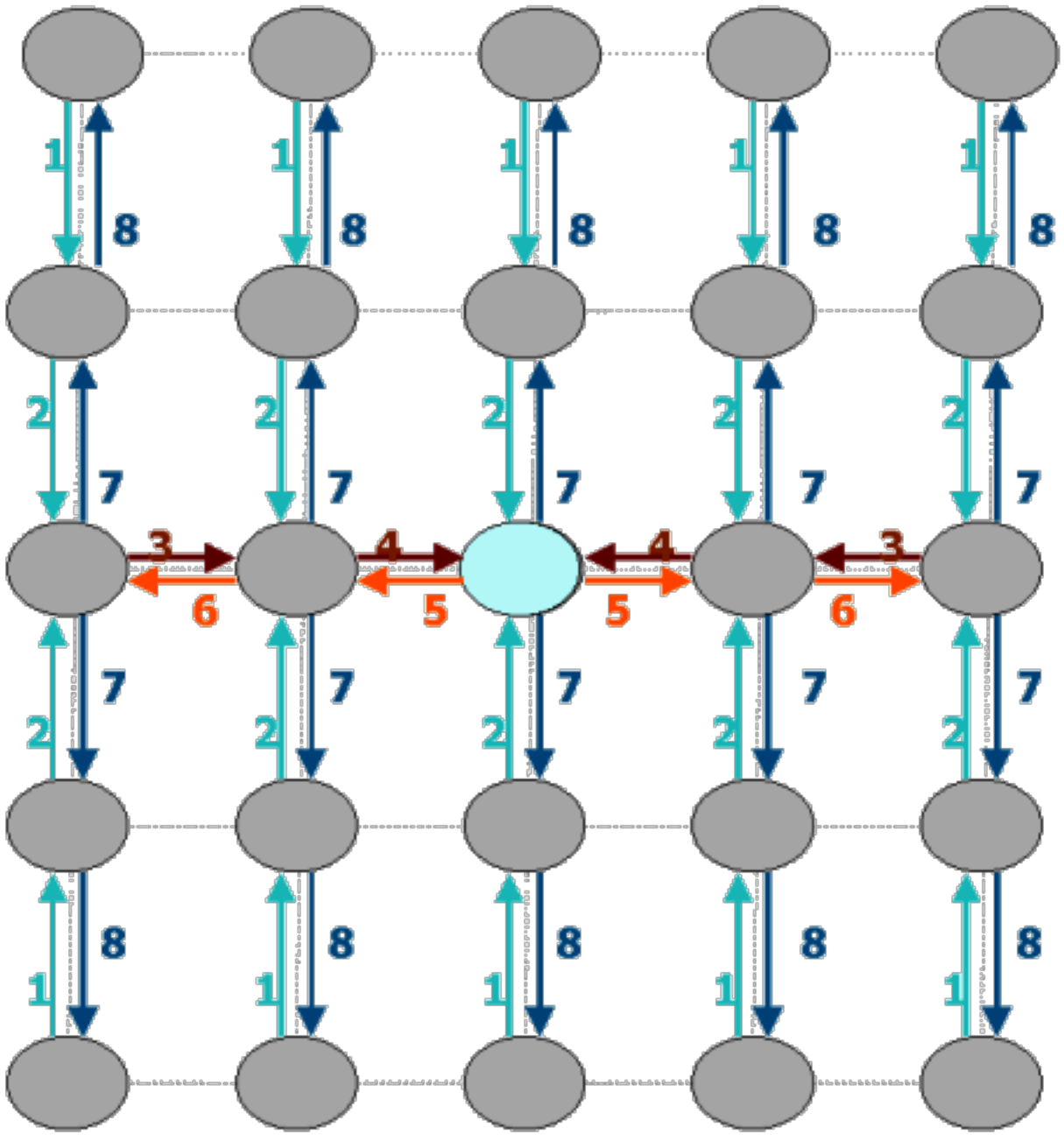
To avoid communication with the diagonal processors message overlapping is applied



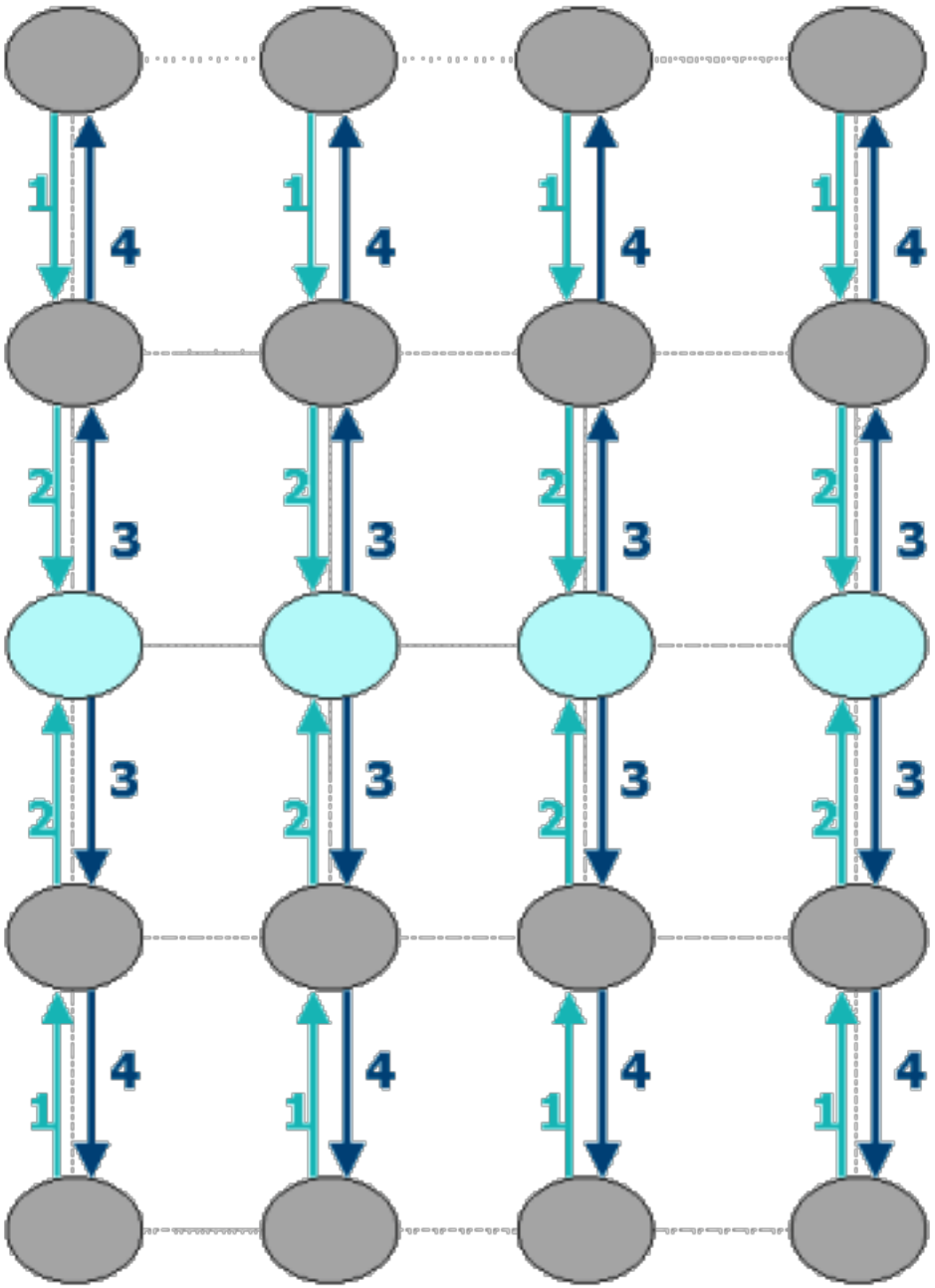
Row Exchange

- Each subdomain consists of kernel points (‘X’) and Halo points (‘O’)
- Local communication is applied once per module
- Communication wherever possible is substituted by Halo point computations.

Global Communication



Mixed Communication



Message Passing

Communication is carried out in two phases

- **First Phase** the column boundaries are exchanged with the left and right neighbour
- **Second Phase** the row boundaries are exchanged with the upper and lower neighbour

The proposed communication scheme consists of four Send calls that can be performed in pairs concurrently, and four blocking Receive calls. The two phases may be performed in any order

```
Send_left    (message1)
Send_right   (message2)
Recv_right   (message1)
Recv_left    (message2)
Send_up      (message3)
Send_down    (message4)
Recv_down    (message3)
Recv_up      (message4)
```

or

```
Send_up      (message1)
Send_down    (message2)
Recv_down    (message1)
Recv_up      (message2)
Send_left    (message3)
Send_right   (message4)
Recv_right   (message3)
Recv_left    (message4)
```

Parallel Implementation Issues

- A communication library has been developed which contains all the machine and interface communication routines
- Code that would execute on both shared and distributed memory architectures via message passing
- A data partitioning scheme is used to allocate the data to the processors
  - For each processor that takes part in the data mapping, special routines have been developed to decide its position on the mesh, its neighbours and the dimensions of its kernel and halo points
- Communication is performed at the beginning of the subprogram that requires it

Parallel Implementation Details

- Data Partitioning: allocating the data to the processors
- Topology: a two dimensional mesh where each processor will communicate with his top, down, left and right adjacent processor
- Processor Identification: numbering the processors from 0 to np-1, and relating them to the coordinates of the position they hold on the two-dimensional mesh topology
- Parallel Output: a postprocessing part which joins the output data of the parallel program in files, such as to produce the appropriate output

Performance

Model Problem: Forecast simulation with 32 layers of 241x321 grid points each (0.125^0 grid spacing)

Simulation time: 12 hours

Number of steps: 960

Data decomposition: rectangular subdomains

Mapping scheme: checkerboard partitioning

Parallel interface: MPI

Target machine: HP-UX SPP-2000 with 64 procs

Target Machines

- HP Exemplar SPP-2000: Four hypernodes with 64 processors PA-RISC 8000-180MHz, 8GB RAM and 1 MB per processor cache

- Parsytec Cce-16: Each node is a Power PC-200 MHz with 128 MB RAM on each node. The internal network is a high network with about 40 MB/sec node to node bandwidth

Future Work

- Load Balancing
  - Static
  - Dynamic
- Irregular data mapping
- Dynamic processor topologies
- Machine dependent optimization
- Inclusion of Dust uptakeTransport
- Coupling Parallel-Eta & Parallel-POM
- Parallelization of the vertical computations
- Study of using implicit numerical schemes in the horizontal mesh