

Mandatory Assignment 3

The Frosty Trap



Frosty Trap

- Successive Over Relaxation
- A very common computational kernel in many scientific applications
- A typical example of grid-communication applications
- Can achieve very good speedup











```
do {
    diff=0.0;
    for i = 1 to n-1
    for j = 1 to n-1 {
        temp = A[i,j]
        A[i,j] = 0.2 * (A[i,j] + A[i+1,j] + A[i-1,j] + A[i,j+1] + A[i,j-1])
        diff = diff + abs( A[i,j] - temp )
     }
} while ( diff>epsilon )
```



Dependency

 \bigcirc \bigcirc ()() () () () ()()()()() () ()()()()()́)) () () () $\left(\right)$ \bigcap () ()→) ົ)◀--() \bigcap \frown () \bigcirc ()()() () ()



Wavefront

 \bigcirc \bigcirc \bigcirc Iteration 4 ()) Iteration 3 ()Iteration 2 \bigcirc () () Iteration 1 \bigcirc () \bigcirc () \bigcirc ()()()()()() \bigcirc \bigcirc \bigcirc ()()() () $\bigcirc \bigcirc$ \bigcirc \bigcirc \bigcirc () ()()



Red'n'Black





Assignment

- Write a parallel version of the sequential application
- Evaluate your solution
 - Include performance graph
 - Disable graphics for this



Decomposition

- Striping
- Tiling
- Cyclic striping
- Iteration based



Practical issues

- Keep the report short
 - Analyses
 - Decomposition
 - Static and dynamic approaches
 - Performance
 - Graph
 - Hand in May 29th 9.15
 - That is **before** the lecture