Mandatory Assignment 1

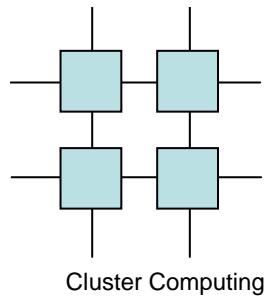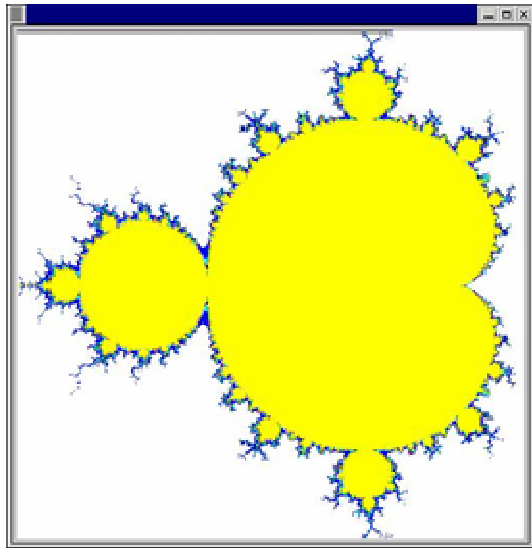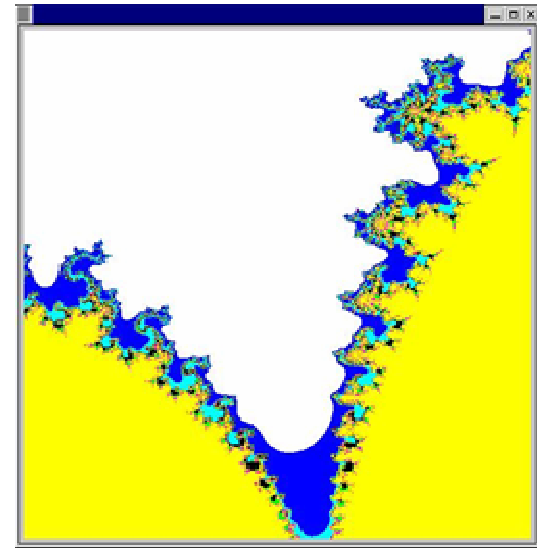## The Road Map

Road Map

Cluster Computing
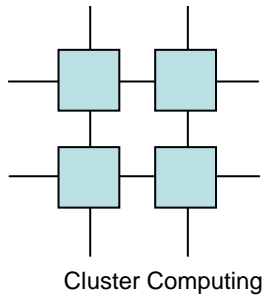
- Fractals are examples of applications of the type we call embarrassingly parallel
- A typical example of an compute intensive application with many independent sub-results
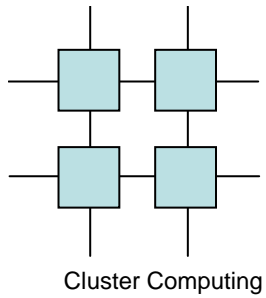- Very simple to write
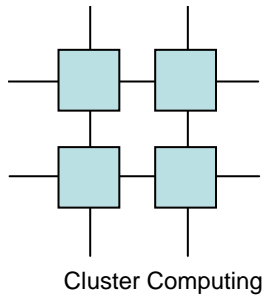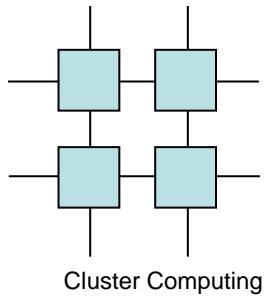- Achieves very good speedup

Road Map

. . .

Process

$$f(x, y, r, s) = f(x, y, r^2 - s^2 + x, 2rs + y)$$

Dependencies?

Assignment

Cluster Computing

- Write two parallel versions of the sequential zooming-roadmap application
  - One with static orchestration
  - One with dynamic orchestration
- Evaluate your solution
 - Include performance graph
  - Disable graphics for this

Decomposition

- Split the task into n frames
- Split the map into n blocks horizontally
- Split the map into n blocks vertically
- Split the map into n grid-blocks
- Take every n^{th} row
- Take every n^{th} column
- Take every n^{th} point

Practical issues

Cluster Computing
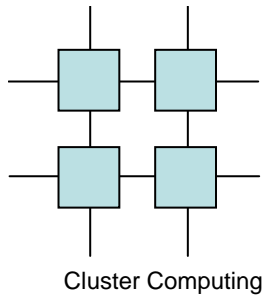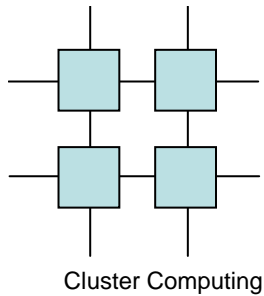
- Keep the report short
  - Analyses
  - Decomposition
  - Static and dynamic approaches
  - Performance
    - Graph

  - Hand in May 10th 9.15
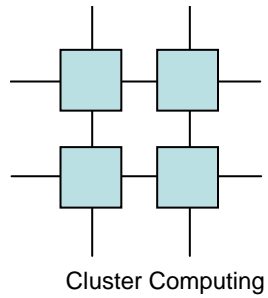   - That is **before** the lecture
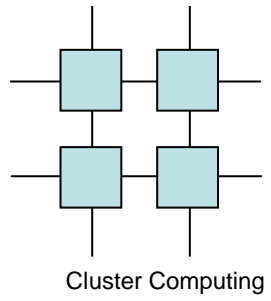
Results and Analyses

Your report card

- **Important is**
  - Accepted/Not Accepted
- **Least important is the following number**
  - It is the CPU utilization you got with the static orchestrated version – this is only used fo rthe competition
- **The comments are usually very short**
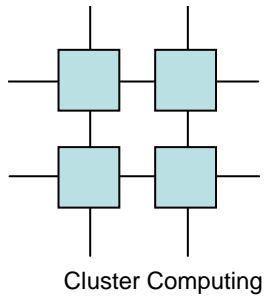 - Fine report, solid report, OK report or specific short comings

A picture is worth 1000 words…

- Illustrate your decomposition with a drawing – that is much easier to communicate with the reader
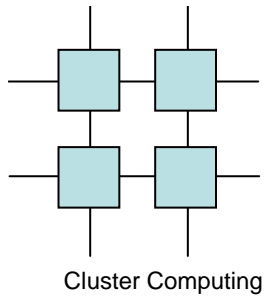
Graphics is slow

- Turning graphics off during measurements makes better and more stable results

Graphs

- Speedup graphs measure CPUs vs Speedup (=improvements over sequential version)
  - Speedup graphs should raise
- Runtime graphs measures CPUs vs total runtime
  - Runtime graphs should fall
- CPU utilization graphs shows Speedup divided with CPUs
  - Utilization graphs should be straight (but tends to all)
- Use the correct notation of your graph!
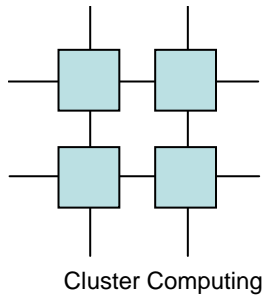
Results

- ## Show your speedup!
  - Not just your runtime
- ## Speedup = t_{seq}/t_{par}

- ## Alternatively show CPU utilization
  - CPU utilization = Speedup/N_{CPU}