

Name: Trevor Buck

ID: 109081318

CSCI 3104, Algorithms  
Explain-It-Back 3

Profs. Grochow & Layer  
Spring 2019, CU-Boulder

You colleagues in the meteorology department need your help scaling their weather prediction capabilities to a wider area. They explain that weather prediction involves dividing the forecast region into a grid, computing a short-term prediction for each cell in the grid based on conditions from weather sensors, then combining those results for region-wide forecast. Right now they start in the top left cell of the grid, then proceed right cell by cell, then row by row, until a prediction has been made in every cell. They then go back over the grid and consolidate the forecasts in a 2x2 cell square starting in the top left and moving right across the grid two cells at a time (i.e., the squares do not overlap). This is then repeated several more times considering ever larger squares, until they have one final prediction. Their department has invested in a new large multi-processor computer to help them make predictions over larger areas, but they do not know how to utilize all of these new processors since the computation of individual cells, or groups of cells, cannot be efficiently computed by multiple CPUs. Help them understand how a new divide-and-conquer strategy may help them better leverage the power of their new hardware without adding extra work to the overall solution.

NOTE: For simplicity, assume that there is no direct processor-to-processor communication. A CPU gathers data stored in a cell or group of cells, runs a program on that data, and writes the result back to the same cell/cells which can then be read by some other processor. There are many real-world strategies for coordinating work among many processors. The one we assume here is similar to what is used in the map-reduce/Hadoop framework.

Congratulations on obtaining the new multi-processor computer! That's awesome! I've worked with a computer like that and it definitely speeds things up. Even more so, if you know how to use it correctly ☺

What I mean by that, is that it is very beneficial to adapt your programs to run better on a multi-processor. Lets take your weather prediction model for example. Right now, it just kinda pounds things out, line for line. But might I suggest a different strategy. Instead of computing one cell at a time, then four, then eight, then sixteen... I would run four individual cells, then combine those into a  $2 \times 2$ . Then do that again, 4 times and make a  $2 \times 2$  thats made of  $2 \times 2$ 's i.e.  $\begin{pmatrix} \boxed{\boxed{\quad}} & \boxed{\boxed{\quad}} \\ \boxed{\boxed{\quad}} & \boxed{\boxed{\quad}} \end{pmatrix}$  As well, you could split the work among the processors (i.e. if you have a  $64 \times 64$  and 4 processors, have P1 do 1-16, P2 do 17-32, P3 do 33-48, and P4 do 49-64)

The goal is to split the work and do more work at each steps to reduce the number of steps taken.