

Name: Michael Rogers

ID: 105667404

CSCI 3104, Algorithms
Explain-It-Back 3

Profs. Grochow & Layer
Spring 2019, CU-Boulder

Your 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.

Name: Michael Rogers

ID: 105667404

CSCI 3104, Algorithms
Explain-It-Back 3

Profs. Grochow & Layer
Spring 2019, CU-Boulder

Hello colleagues in the meteorology department,

It's great to hear that the equipment in your department has been upgraded. It is going to be a huge help in making your predictions more accurate and gathering data faster. With more efficient and more powerful hardware, you must also upgrade your algorithm and how you process this data at the software level. While the process you described to me is a logical way of working through all the data, it will not be nearly as efficient or as fast as other methods. Currently, you are suggesting what we call a linear algorithm in the computer science field. We call it this because it goes through each data point in a line and then goes back to the beginning of the line and repeats the process. This is a very natural idea of how to collect this data, however, I believe if we use an algorithm that is non-linear (meaning break the data into different subsets) we can much more efficiently get through this data. I suggest we use something called the divide and conquer strategy to collect this data. The divide and conquer strategy uses that non-linear algorithm that I mentioned earlier. Let me elaborate, if we divide up the data into different subsets and assign them to different CPUs, then each CPU can work together in getting through the data. We will assign each CPU to have a lighter workload, each will collect its data, and then we will combine the data of each CPU at the end. We will start with the individual weather sensors and divide them up amongst the CPUs. Once we have collected the smaller scale data, each CPU can begin to move up in scale to a larger region until we have worked through the entire region. Next, we can combine this data to get our result which is the large region prediction that we are after. A good way to think about it is this: if an army wanted to take control of an area, they wouldn't come at the enemy in a giant mass of soldiers. Instead, they would split up and take control of smaller areas, then combine those areas until they had conquered the territory that they set out to take control of. This strategy should significantly increase the speed of your program as well as the efficiency. I would be happy to help you implement this program, if you have any questions feel free to contact me.

Good Luck,
A Computer Scientist You Know