Mentoring through the Spark Program by David Richter

For six weeks this past summer, I had the wonderful opportunity to mentor Edward Montoya, a recent graduate of Adams City High School in Commerce City, CO, through the pre-college internship program provided by Spark (spark.ucar.edu). Mentoring was an excellent experience which I recommend anyone at NCAR to consider, and I hope that a brief description of these six weeks will illustrate how beneficial this opportunity can be for both intern and mentor.

I had originally anticipated being a mentor through the SOARS program, which is similar in nature to the pre-college internship program but involves working with students making the undergraduate to graduate transition rather than the high school to college transition. In all honesty, I was at first hesitant to work with students who had not yet had any college experience. It turns out, however, that more help was needed in the pre-college program, so my plans changed. To explain my initial hesitancy, I challenge the reader to recall their level of knowledge at the time of their high school graduation - if you are anything like me, that took longer than expected. It was difficult to recall where and when various computer skills, writing abilities, etc., were picked up along the way, and I was concerned that I might mistakenly overwhelm the student. Remembering these things was my first task as I began considering project ideas for my intern.

Spark did a wonderful job in pairing me with Edward, as he is interested in computers and engineering (as opposed to meteorology or climate). In fact, this fall he will start in the Electrical, Computer, and Energy (ECE) Engineering department at CU. My background is in mechanical engineering, but my work here involves large-scale computer simulations of turbulence in the atmosphere. Therefore, I was optimistic that the various computational aspects of my work would appeal to him, resulting in a project that we could both enjoy and learn from. This was...
Mentoring (continued)

most certainly the case. Edward is a very bright, soft-spoken student, interested in a wide variety of topics. He was involved with music, soccer, and even marine biology during high school, and had taken a large number of courses for college credit. Despite not having any formal background in computer science or engineering, he repeatedly surprised me with his ability to quickly become comfortable with things like navigating a Unix system, understanding basics of code parallelization, and running jobs on Bluefire. He even converted his personal computer to Linux halfway through the summer. Edward was also very conscientious, each week putting in several hours above the typical amount for pre-college students throughout the internship.

After being completely overwhelmed with information the first day here, he immediately got to work on his project. He began by writing a few simple computer programs in Fortran and C, so that he was at least somewhat familiar with the code he would be using during his internship. This code, the NCAR Large Eddy Simulation Code, was developed in the boundary layer turbulence group of MMM, and it is the primary tool of my postdoctoral research. It was originally intended to perform large eddy simulations of the atmospheric boundary layer, but I have recently altered it in a way that allows for particle-laden, fine-scale turbulent flows to be simulated and analyzed. As with any of the large-scale codes developed at NCAR or elsewhere, the ability to efficiently run on hundreds or even thousands of processors is essential to fully utilizing the capabilities of Bluefire and other supercomputers. These supercomputers consist of thousands of individual CPUs linked together. Properly (i.e., evenly) distributing the total amount of work being done in a particular code across all of these CPUs is sometimes complicated but necessary. This even distribution of computational work over all of the requested CPUs was the basis for Edward’s project: Finding an optimal configuration of the code, which would lead to the fastest speed possible for a given number of CPUs.

In the code, turbulent flow inside of a rectangular, three-dimensional volume is being computed, while at the same time, small particles are being transported by the turbulent motions. To distribute this problem over a large number of processors (call this number N), the 3D volume is split up into N sub-volumes, each belonging to (and being solved by) a single CPU. The question at this point becomes: What is the ideal size and shape of these sub-volumes to maximize the speed of the code? Imagine a square divided into 64 segments, kind of like a giant tic-tac-toe board. There are many ways of arranging these segments. You could choose to distribute evenly in both directions, with 8 divisions in one direction and 8 in the other, resulting in 64 smaller squares. Or you could choose 2 segments in one direction with 32 in the other, resulting in 64 thin, tall columns. In the simulation code, these segments represent the volume being solved by a particular processor, and each shape may lead to a different runtime speed due to factors such as communication between CPUs. During his internship, Edward tested all combinations of processor configuration and ultimately determined which took the shortest and which took the longest time to run.

Mentoring a young student was, as I mentioned above, an experience that I hope anyone would consider. It was a pleasure working with Edward on a daily basis, and watching him learn and become excited about high-performance computing was well worth the time commitment. Guiding him through this project was also beneficial in more tangible ways, clarifying to myself the different considerations of how to optimize the code, and at the end of the summer I now have a concrete idea of how to maximize the speed of my research simulations. I hope that Edward now has a great head start on his path towards a computer engineering degree, and I look forward to keeping in touch with him as he proceeds.

David Richter is an ASP Postdoctoral Fellow working in MMM. He joined ASP in 2011 after obtaining his degree in Mechanical Engineering from Stanford University.

Mentoring is a brain to pick, an ear to listen, and a push in the right direction.

~John Crosby