Computer Science Colloquium

Making Parallel Programming Easier using Patterned Programming Techniques and Tools

Clayton Ferner
University of North Carolina at Wilmington

Thursday, October 2, 2014
12:30 – 1:30 pm
Petty Building, Room 313

Abstract: With multi-core processors, GPUs, and clusters made of commodity computers, parallel computer systems are now readily available to all. Even with the availability of parallel systems, very little software is designed or written to take advantage of parallel architectures. There is a great need to train software developers in how to make use of parallel processing. Indeed, the ACM/IEEE Computer Science Curriculum 2013 considers parallel and distributed computing to be a knowledge area and recommends that it be required for all CS curricula. There is also a trend to introduce parallel computing into the lower level computer science courses.

The challenge with parallel computing is that writing software for multiple concurrent streams of execution is fraught with difficulties, such as deadlocks and race conditions. A recent trend in parallel computing is using patterns to express parallel algorithms. We have developed several tools that will allow the programmer to express a parallel algorithm using patterns and will automatically generate the parallel software. The advantage of this is that the programmer is now thinking about the problem at a higher-level of abstraction.

In this talk, I will discuss patterns and introduce one of the tools that we have developed. I will talk about using these tools in the classroom to teach parallel computing.

About the Speaker: Dr. Clayton Ferner is a professor of Computer Science at the University of North Carolina Wilmington (UNCW) and Director of the Master of Science program in Computer Science and Information Systems. He has been a faculty member at UNCW for 15 years and taught courses on parallel computing, grid computing, programming languages, compiler construction, operating systems, networks, and data structures. He received his Ph.D. in mathematics and computer science from the University of Denver in 1997.