



Events

[Current Events](#)[Lectures ▾](#)[Events Archive ▾](#)

Special Guest Lectures

Computer Performance Analysis and the Pi Theorem**Robert W. Numrich, Minnesota Supercomputing Institute**

University of Minnesota, Minneapolis

Johnston Hall 338

January 18, 2007 - 09:30 am

Abstract:

We apply dimensional analysis to computer performance analysis. Dimensional analysis is the study of self-similarity, which means that a phenomenon reproduces itself on different time and/or space scales (Barenblatt). Self-similarity laws are scaling laws, and scaling never appears by accident (Barenblatt). The fundamental tool of dimensional analysis is the Pi Theorem. The Pi Theorem is a mathematical statement of the equivalence of physical systems, an idea with a long history dating back at least to Newton. It says that physical systems don't care what units we use to measure them. Dimensional analysis is valid for computer systems as much as it is for other physical systems, but it has been used only sparingly for computer performance analysis. We use the Pi Theorem to obtain a new scaling formula for the Linpack benchmark, to obtain scaling formulas that explain cache-miss ratios, and to obtain self-similarity surfaces for parallel matrix multiplication.

Speaker's Bio:

Bob Numrich is a Senior Research Associate at the Minnesota Supercomputing Institute, University of Minnesota, Minneapolis. His research interests include parallel architectures, parallel programming languages, and parallel numerical algorithms. He also studies computer performance analysis and the development of theoretical models that yield self-similarity relationships between systems. Previous to his position at the University of Minnesota, he was Principal Scientist at Cray Research where he worked on the Cray-2 and Cray-3 architectures and was a member of the core development teams for the Cray-T3D and Cray-T3E. He invented the one-sided parallel programming model that became the SHMEM Library, and he is the principal author of the Co-Array Fortran programming model.

Refreshments will be served.**This lecture has a reception.**