Towards a Science of Parallel Programming
Keshav Pingali, University of Texas at Austin

When parallel programming started in the 70's and 80's, it was mostly art: languages such as functional and logic programming languages were designed and appreciated mainly for their elegance and beauty. More recently, parallel programming has become engineering: imperative languages like FORTRAN and C++ have been extended with parallel constructs, and we now spend our time benchmarking and tweaking large programs no one understands to obtain performance improvements of 5-10%. In spite of all this activity, we have few insights into how to write parallel programs to exploit the performance potential of multicore processors.

In this talk, I will argue that these problems arise largely from the limitations of the program-centric abstractions like dependence graphs that we currently use to think about parallelism. I will then propose a novel data-centric abstraction called the operator formulation of algorithms, which reveals that a generalized form of data-parallelism called amorphous data-parallelism is ubiquitous in diverse applications ranging from mesh generation/refinement/partitioning to SAT solvers, maxflow algorithms, stencil computations and event-driven simulation. I will also show that the operator formulation can be used to perform a structural analysis of algorithms that can be exploited for efficient implementations of these algorithms. Finally, I will describe a system based on these ideas called Galois for programming multicore processors.