

Why has compilers failed to help parallel programmers

As we are moving into an era of parallel programming, it is clear that current best practices put all the burden of building correct and high performance programs on the programmer. There is absolutely little help from existing languages and compilers. While we had decades of research and implementation of high performance, high level languages with compiler support, none of them were widely adopted. In this talk I will examine what it takes to build a successful compiler. I will introduce five requirements for wide adaptation of a compilation technique – robustness, scalability, simplicity, portability and effectiveness. Building a successful compiler largely depends on having the right level of language abstraction. If the abstraction is too high, it can overburden the compiler. On the other hand, if it is too low, the necessary choices are not available to the compiler without heroic analysis. I will discuss what I believe is the right level of abstraction for a high performance, high level language.