Code Optimizations and Their Automation

Alain Darte
CNRS research scientist, ENS Lyon, Laboratoire de l'Informatique du Parallélisme, Compsys team

With highly-pipelined vector processors, hardware accelerators (GPU, FPGA), deeper memory hierarchies, and heterogeneous designs becoming mainstream, programmability, portability, and productivity are now important facets to consider on the way to performance. In this context, I will try to illustrate how compiler research, through language developments and automation of code analysis and optimizations, addresses these concerns and why the gap between what compilers can do and what HPC users hope they could do is still very large.

Cost models (for both the users and the compilers), exchange between compiler and user (application knowledge and optimization reporting), and limits of code analysis, remain serious issues. Nevertheless cost models (such as roofline and ECM), communications (such as automatic offloading), locality optimizations (such as tiling), language design (of various kinds) still make regular progress in these directions.