Automatic Parallelization in GCC: for Research and for Real (Keynote Talk)

Albert Cohen

Automatic Parallelization in GCC: for Research and for Real

Albert Cohen
INRIA

ABSTRACT

Portability of performance has been the underlying assumption for the vast majority of software developments. Micro-architectures and run-time systems have been designed to hide the parallelism and non-uniformity of the hardware, but with diminishing returns in performance and poor power efficiency. Despite decades of successes with optimizing compilers, the complexity of modern hardware has incrementally destroyed this dream for most developers. Every day, more programmers are forced to resort to platform-specific optimizations, committing early on specific parallel implementations. This is a dramatic regression.

This talk argues that portability of performance has to be our ambition. Production compilers and run-time systems must address this challenge, rejuvenating the research and development of automatic parallelization techniques for general-purpose applications. We will report on recent results and ongoing attempts to extend the capabilities of the GNU Compiler Collection (GCC), including annotation-based parallelization, data-flow and stream programming, polyhedral compilation, iterative and machine-learning compilation. We will discuss how these attempts try to break down or circumvent the effectiveness limitations of state-of-the-art parallelizing compilers. Eventually, we will sketch promising directions, and advocate for building research prototypes as components of real-world compiler frameworks, together with developer communities.