Abstract

Unleashing the full potential of modern CPUs requires making good use of the architecture's parallelism in terms of both multi-core and, ever more importantly, SIMD. In this talk, we first introduce IVL, an experimental SPMD compiler in which the user writes a scalar program that the compiler then maps to SIMD by running a separate instance of this program in each SIMD lane. IVL uses a scalar C(++)-like syntax with some simple additional keywords to express parallelism and data layouts, and currently has back-ends for SSE, AVX, and MIC/Knights*, as well as support for "device offload" where host and device code run on different devices (e.g., on a separate KNF card). To demonstrate the power of this framework we then briefly summarize results from (R)IVL, a complete real-time-high-quality renderer in which all rendering code has been written exclusively in IVL, and that is fully portable between SSE, AVX, and MIC.