

Vector Parallelism on Multi-Core Processors

Thursday 25 July 2019 08:30 (2 hours)

All modern CPUs boost their performance through vector processing units (VPUs). VPUs are activated through special SIMD instructions that load multiple numbers into extra-wide registers and operate on them simultaneously. Intel's latest processors feature a plethora of 512-bit vector registers, as well as 1 or 2 VPUs per core, each of which can operate on 16 floats or 8 doubles in every cycle. Typically these SIMD gains are achieved not by the programmer directly, but by (a) the compiler through automatic vectorization of simple loops in the source code, or (b) function calls to highly vectorized performance libraries. Either way, vectorization is a significant component of parallel performance on CPUs, and to maximize performance, it is important to consider how well one's code is vectorized.

In the first part of the presentation, we take a look at vector hardware, then turn to simple code examples that illustrate how compiler-generated vectorization works and the crucial role of memory bandwidth in limiting the vector processing rate. What does it really take to reach the processor's nominal peak of floating-point performance? What can we learn from things like roofline analysis and compiler optimization reports? And what can a developer do to help out the compiler?

In the second part, we consider how a physics application may be restructured to take better advantage of vectorization. In particular, we focus on the Matriplex concept that is used to implement parallel Kalman filtering in our collaboration's particle tracking R&D project. Drastic changes to data structures and loops were required to help the compiler find the SIMD opportunities in the algorithm. In certain places, vector operations were even enforced through calls to intrinsic functions. We examine a suite of test codes that helped to isolate the performance impact of the Matriplex class on the basic Kalman filter operations.

Presenter: LANTZ, Steven R (Cornell University (US))