



Contribution ID: 2

Type: not specified

II - Template Metaprogramming for Massively Parallel Scientific Computing - Vectorization with Expression Templates

Tuesday, 1 March 2016 11:30 (1 hour)

Large scale scientific computing raises questions on different levels ranging from the formulation of the problems to the choice of the best algorithms and their implementation for a specific platform. There are similarities in these different topics that can be exploited by modern-style C++ template metaprogramming techniques to produce readable, maintainable and generic code. Traditional low-level code tend to be fast but platform-dependent, and it obfuscates the meaning of the algorithm. On the other hand, object-oriented approach is nice to read, but may come with an inherent performance penalty.

These lectures aim to present the basics of the Expression Template (ET) idiom which allows us to keep the object-oriented approach without sacrificing performance. We will in particular show to enhance ET to include SIMD vectorization. We will then introduce techniques for abstracting iteration, and introduce thread-level parallelism for use in heavy data-centric loads. We will show to apply these methods in a way which keeps the “front end” code very readable.

LECTURE 2

In this lecture, we will have a closer look at the opportunities for implementing SIMD vectorisation through the Expression Template idiom. We will see how it can create a layer of separation between the algorithm, and the low-level implementation. We will use the C++ template mechanisms to accommodate our program so that the algorithm itself doesn't need to explicitly specify SIMD-related types alignment, or operations. We will also explore how our memory data structure layout affects SIMD performance in different workloads, and introduce methods which improve performance in specific cases.

Presenter: VYSKOČIL, Jiří (Czech Technical University in Prague)