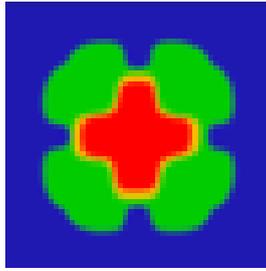


Sixth deal.II Users and Developers Workshop



Contribution ID: 2

Type: **Keynote presentations**

Supports for GPU in deal.II

Wednesday, July 25, 2018 9:00 AM (45 minutes)

GPUs are extensively used in supercomputers both in Europe with Piz Daint and in the United States with Titan, Sierra, and Summit. The main reason is that they offer great floating points performance for a given power envelope. GPUs are also the de facto standard architecture for neural networks whose usage has been increasing quite rapidly in the last few years. It is therefore very advantageous if other applications can use the same infrastructure as the one used for AI. However, programming on GPU is not the same as programming on CPU. GPUs offer a lot more parallelism but GPU cores are a lot less powerful than CPU cores.

In this talk, we will discuss how to use deal.II on GPUs for matrix-based and matrix-free applications. For matrix-based applications, we rely on cuSPARSE and cuSOLVER for operations on sparse matrices and for some solvers. Matrix-free methods rely on the fact that many modern codes are using Krylov methods as their solver of choice. These solvers do not require the matrix of the system but only the result of a sparse matrix-vector multiplication. This operation is memory bound both on CPU and GPU. Matrix-free methods reduce drastically the memory requirement and thus, allow for a better utilization of the hardware. Moreover, the extensive use of tensor products make these methods extremely interesting for high-order finite elements. High-order finite elements are attractive not only because of the convergence properties but also because they introduce structure even on unstructured meshes. Finally, the lesser memory requirement of matrix-free algorithms is very advantageous for GPUs since they do not have a lot of memory per core. We will show some results and discuss our path forward.

Primary author: TURCK SIN, Bruno (ORNL)

Presenter: TURCK SIN, Bruno (ORNL)

Session Classification: Keynotes

Track Classification: Developers' track