Enabling the productive development of efficient partitioned-address-space parallel applications from global-address-space abstractions

P. Sadayappan (Ohio State University)

The dominant parallel programming model today is MPI, which imposes a partitioned view of the address space across the parallel processes. For most MPI-based applications, the core algorithms are not very different from those that would have been used for a sequential implementation of the computation. The available parallelism in the algorithms is also generally not difficult to identify. However, a very significant effort is required in translating the abstract parallel algorithms into a concrete MPI program because of the need to create a partitioned representation of the data structures, and the orchestration of explicit communication between the processes for needed portions of the data that are not located on the data partition associated with each process.

The process of developing message-passing applications “from scratch” is so tedious and error-prone that most large-scale MPI-based applications first develop a layered set of abstractions to create high-level global views for key data structures, where all the message passing needed to move data between processes is implemented in the lower layers and thus hidden at the higher layers. The development of complex models is then done primarily using the high-level abstractions. For example, the USQCD software suite provides such a layered abstraction for multi-dimensional lattices and high-level operations such as shifts on lattices, which abstracts away the low level partitioned view of the lattice among different nodes and the needed message passing to implement the shifts. Similarly, climate codes create their own layers of abstraction to provide a high-level global view of discretized grids that are represented in a partitioned manner at the lower levels.

The current practice of creating layered abstractions for each application domain or application suite is wasteful since there is little reuse of the software infrastructure across domains and even across independently developed applications within a domain. Therefore it is of interest to explore approaches to build common infrastructure that can enable the effective development of efficient partitioned-address-space parallel applications from global-address-space abstractions.

There are a number of research efforts that are focused on the use of global-address-space or PGAS models and abstractions. But what is the extent of use by DOE application scientists? What are the challenges and shortcomings that are inhibiting greater use?

It could be of interest to organize a session focused on this topic, bringing together researchers from the programming-language/compiler/runtime research community and application scientists to assess recent progress and to brainstorm on ways of maximizing future progress.